Pearson BTEC National Computing

BTEC

Student Book

David Atkinson-Beaumont
Tim Cook
Mark Fishpool
Alan Jarvis
Richard McGill
Jenny Phillips

Pearson

Published by Pearson Education Limited, 80 Strand, London, WC2R 0RL.

www.pearsonschoolsandfecolleges.co.uk

Copies of official specifications for all Edexcel qualifications may be found on the website: www.edexcel.com

Edited by Julie Bond
Designed by Andy Magee
Typeset by Tech-Set Ltd, Gateshead

Illustrated by Tech-Set Ltd, Gateshead

Picture research by Alison Prior

First published 2016

19 18 17 16
10 9 8 7 6 5 4 3 2 1

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 978 1 292 27717 2

Acknowledgements
We would like to give a special thanks to Alan Jarvis for his invaluable help, expertise and advice in the development of this content.

The publisher would like to thank the following for their kind permission to reproduce their photographs:

(Key: b-bottom; c-centre; l-left; r-right; t-top)

123RF.com: 452, asife 389 (Avatar), rosipro 362; **Alamy Images:** Chris Cooper-Smith 207, Ian Dagnall 545b, Kurt Miller 448, Neil Newton 186, Robert Kneschke 363; **Getty Images:** alvarez 491, Bloomberg 411l, Buero Monaco 276, Colin Anderson 445, 463r, Corbis 156, kprojekt 112l, Thomas Trutschel 407, vgajic 213; **NASA:** Scott Kelly 135; **Shutterstock.com:** Africa Studio 5tl, Aitormmfoto 5bl, Ana Bokan 444, Andrey_Popov 507bl, arek_malang 406, charnsitr 452r, Darrin Henry 507tr, Diego Cervo 63, Eldad Carin 555, Ilike 507br, iurii 463l, John T Takai 433, Lightspring 301, michaeljung 537, Nadino 443, Olga Zelenskaya 507cl, paul prescott 578, Pieter Beens 545, pkchai 492, racorn 405, 577, Rawpixel.com 127, Roger Jegg - Fotodesign-Jegg.de 538, sheff 507tl, The Turtle Factory 450, tusharkoley 389 (Background), Umberto Shtanzman 69, Valentin Valkov 557, Wavebreakmedia 452l, Zern Liew 544, ZouZou 391, Zurijeeta 361; **5c Shutterstock**: Sam Chadwick/Shutterstock**; 121 Shutterstock:** Foto Andalucia/Shutterstock**; 411r Shutterstock:** Christian Chan/Shutterstock**; 449 Shutterstock:** Rido/Shutterstock**; 472 Shutterstock:** Jacob Lund/Shutterstock**; 493 Shutterstock:** Stan Fisher/Shutterstock**; 539 123RF:** scanrail/123RF**; 569 Shutterstock:** Andrey_Popov/Shutterstock;

All other images © Pearson Education

The authors and publisher would like to thank the following individuals and organisations for their approval and permission to reproduce their materials:

Screenshots
Microsoft product screenshots reprinted with permission from Microsoft Corporation; Screenshot on page 370 from www.facebook.com, Facebook; Screenshots on page 371, page 381, page 382 from www.google.com, Google, Google and the Google logo are registered trademarks of Google Inc., used with permission; Screenshot on page 374 from twitter.com, Twitter, Twitter, Tweet and Twitter Bird Logo are trademarks of Twitter, Inc. or its affiliates.; Screenshots on page 381, page 382, page 499 from google.com, Google, Google and the Google logo are registered trademarks of Google Inc., used with permission; Screenshots on page 387, page 388, page 389, page 390, page 391, page 391, page 399, page 400, page 400 from facebook.com, Facebook; Screenshots on page 395, page 397, page 397, page 397, page 396 from hootsuite.com; Screenshot on page 398 from facebook.com, Facebook Inc., Facebook is a trademark of Facebook, Inc.; Screenshots on page 458, page 476, page 478, page 479, page 479, page 481, page 481, page 482, page 483, page 486 from unrealengine.com, Epic Games Inc.; Screenshot on page 461 from playdead.com, Playdead; Screenshot on page 516 from Pearson Education PLC; Screenshot on page 518 from boston.ac.uk, Boston College; Screenshot on page 533 from http://www.audacityteam.org/, Dominc Mazzoni, Audacity® licensed under the Creative Commons Attribution license version 3.0 at http://creativecommons.org/licenses/by/3.0/; Screenshots on page 547, page 557, page 558, page 558, page 568, page 569, page 569, page 570 from http://developer.android.com/tools/studio/index.html, Google, Google and the Google logo are registered trademarks of Google Inc., used with permission

Contents

How to use this book

Welcome to your BTEC National Computing course!

The BTEC National in Computing qualification is a vocational qualification that will help prepare you for a range of careers in one of the most exciting and dynamic subjects. You may be thinking of pursuing a career as a programmer, games or web developer or as a project manager with a focus on software development. A BTEC National in Computing will expose you to a range of computing topics that will enhance your options for employment or for further study in higher education.

You will develop a common core of computing knowledge and study areas such as the fundamentals of computer science, the relationship between hardware and software that form an IT system, and issues related to IT systems security. You will also study a range of optional units such as designing user interfaces, mobile apps development and system analysis and design.

The BTEC National in Computing is well suited to people who want to follow a career in software development as it provides a grounding in the most essential topics relating to this area and also allows the development of specialist skills in areas such as web, games and mobile apps development.

How your BTEC is structured

Your BTEC National is divided into **mandatory units** (the ones you must do) and **optional units** (the ones you can choose to do).

The number of mandatory and optional units will vary depending on the type of BTEC National you are doing. This book supports all the mandatory units and the optional units to allow you to complete the:

- Extended Certificate, for which units 1 (Principles of Computer Science), unit 2 (Fundamentals of Computer Systems) and unit 7 (IT Systems Security and Encryption) are mandatory.
- Foundation Diploma. The mandatory units for this size of qualification are the same as for the Extended Certificate plus unit 8, (The Business Applications of Social Media).
- Extended Diploma. The mandatory units for this size of qualification are the same as for the Foundation Diploma plus unit 3 (Planning and Management of Computing Projects), unit 4 (Software design and development project) and unit 9 (The Impact of Computing).

If you are completing a 'Tech Bacc' qualification you can also study the Diploma version of the BTEC National. The Diploma is available in four different pathways, focussed on particular careers:

- Computer Science
- Creative Industries
- Computer Systems and Network Support
- Business Information Systems.

Each pathway has a different range of Mandatory and option units, although all have units 1, 3 and 4 as mandatory.

Your learning experience

You may not realise it but you are always learning. Your educational and life experiences are constantly shaping your ideas and thinking, and how you view and engage with the world around you.

You are the person most responsible for your own learning experience so you must understand what you are learning, why you are learning it and why it is important both to your course and to your personal development.

Your learning can be seen as a journey with four phases.

Phase 1	Phase 2	Phase 3	Phase 4
You are introduced to a topic or concept and you start to develop an awareness of what learning is required.	You explore the topic or concept through different methods (e.g. research, questioning, analysis, deep thinking, critical evaluation) and form your own understanding.	You apply your knowledge and skills to a task designed to test your understanding.	You reflect on your learning, evaluate your efforts, identify gaps in your knowledge and look for ways to improve.

During each phase, you will use different learning strategies to secure the core knowledge and skills you need.

This student book has been written using similar learning principles, strategies and tools. It has been designed to support your learning journey, to give you control over your own learning, and to equip you with the knowledge, understanding and tools you need to be successful in your future studies or career.

Features of this book

This student book contains many different features. They are there to help you learn about key topics in different ways and understand them from multiple perspectives. Together, these features:

- explain what your learning is about
- help you to build your knowledge
- help you to understand how to succeed in your assessment
- help you to reflect on and evaluate your learning
- help you to link your learning to the workplace.

Each individual feature has a specific purpose, designed to support important learning strategies. For example, some features will:

- encourage you to question assumptions about what you are learning
- help you to think beyond what you are reading about
- help you to make connections between different areas of your learning and across units
- draw comparisons between your own learning and real-world workplace environments
- help you to develop some of the important skills you will need for the workplace, including team work, effective communication and problem solving.

Features that explain what your learning is about

Getting to know your unit

This section introduces the unit and explains how you will be assessed. It gives an overview of what will be covered and will help you to understand why you are doing the things you are asked to do in this unit.

Getting started

This is designed to get you thinking about the unit and what it involves. This feature will also help you to identify what you may already know about some of the topics in the unit and act as a starting point for understanding the skills and knowledge you will need to develop to complete the unit.

Features that help you to build your knowledge

Research

This asks you to research a topic in greater depth. These features will help to expand your understanding of a topic and develop your research and investigation skills. All of this will be invaluable for your future progression, both professionally and academically.

Worked example

Worked examples show the process you need to follow to solve a problem, such as a maths or science equation, or the process for writing a letter or memo. They will help you to develop your understanding and your numeracy and literacy skills.

Theory into practice

In this feature, you will be asked to consider the workplace or industry implications of a topic or concept from the unit. This will help you to understand the relevance of your current learning and the ways in which it may affect a future career in your chosen sector.

Discussion

Discussion features encourage you to talk to other students about a topic, working together to increase your understanding of the topic and to understand other people's perspectives on an issue. These features will also help to build your teamworking skills, which will be invaluable in your future professional and academic career.

Safety tip

These tips give advice about health and safety when working on the unit. They will help to build your knowledge about best practice in the workplace, as well as making sure that you stay safe.

Key terms

Concise and simple definitions are provided for key words, phrases and concepts, giving you, at a glance, a clear understanding of the key ideas in each unit.

Link

Link features show any links between content in different units or within the same unit helping you to identify knowledge you have learned elsewhere that will help you to achieve the requirements of the unit. Remember, although your BTEC National is made up of several units, there are common themes that are explored from different perspectives across the whole of your course.

Further reading and resources

This feature lists other resources – such as books, journals, articles or websites – that you can use to expand your knowledge of the unit content. This is a good opportunity for you to take responsibility for your own learning and prepare for research tasks you may need to complete academically or professionally.

Features connected to your assessment

Your course is made up of mandatory and optional units. There are two different types of mandatory unit:

- externally assessed
- internally assessed.

The features that support you in preparing for assessment are below. But first, what is the difference between these two different types of unit?

Externally assessed units

These units will give you the opportunity to demonstrate your knowledge and understanding, or your skills, in a direct way. For these units you will complete a task, set directly by Pearson, in controlled conditions. This could take the form of an exam or it could be another type of task. You may have the opportunity to prepare in advance, to research and make notes about a topic, which can be used when completing the assessment.

Internally assessed units

Most of your units will be internally assessed and will involve you completing a series of assignments, set and marked by your tutor. The assignments you complete will allow you to demonstrate your learning in a number of different ways, from a written report to a presentation to a video recording and observation statements of you completing a practical task. Whatever the method, you will need to make sure you have clear evidence of what you have achieved and how you did it.

Assessment practice

These features give you the opportunity to practise some of the skills you will need during the unit assessment. They do not fully reflect the actual assessment tasks but will help you to prepare for them

Plan – Do - Review

You will also find handy advice on how to plan, complete and evaluate your work. This is designed to get you thinking about the best way to complete your work and to build your skills and experience before doing the actual assessment. These questions will prompt you to think about the way you work and why particular tasks are relevant

Getting ready for assessment

For internally assessed units, this is a case study of a BTEC National student, talking about how they planned and carried out their assignment work and what they would do differently if they were to do it again. It will give you advice on preparing for your internal assessments, including Think about it points for you to consider for your own development.

Getting ready for assessment

This section will help you to prepare for external assessment. It gives practical advice on preparing for and sitting exams or a set task. It provides a series of sample answers for the types of question you will need to answer in your external assessment, including guidance on the good points of these answers and ways in which they could be improved.

Features to help you reflect on and evaluate your learning

PAUSE POINT	Pause Points appear regularly throughout the book and provide opportunities to review and reflect on your learning. The ability to reflect on your own performance is a key skill you will need to develop and use throughout your life, and will be essential whatever your future plans are.
Hint Extend	These sections give you suggestions to help cement your knowledge and indicate other areas you can look at to expand it.

Features which link your learning with the workplace

Case study

Case studies throughout the book will allow you to apply the learning and knowledge from the unit to a scenario from the workplace or industry. Case studies include questions to help you consider the wider context of a topic. They show how the course content is reflected in the real world and help you to build familiarity with issues you may find in a real-world workplace.

THINK FUTURE

This is a case study in which someone working in the industry talks about their job role and the skills they need. The section suggests ways for you to develop the employability skills and experiences you will need to be successful in a career in your chosen sector. This will help you to identify what you could do, inside and outside your BTEC National studies, to build up your employability skills.

About the authors

David Atkinson-Beaumont

David has worked in Games and Computing education for over ten years following a career as a programmer and IT Consultant. He currently manages Truro and Penwith College's Vocational Arts and Community department based in Penzance, looking after courses from Level 1 to Level 6 and has worked for Pearson as a course writer, textbook contributor and external verifier.

Tim Cook

Tim is a Team Leader for QCF and NQF BTEC Computing qualifications, and has worked for Pearson in a variety of consultancy roles for over eight years. He has contributed to writing several computing specifications, including the new NQF 2016 BTEC Level 3 Computing qualification. Alongside his work for Pearson, he teaches BTEC computing qualifications at Boston College. He has done this for over fifteen years, and has delivered computing based units from Level 1 up to HND.

Mark Fishpool

Mark has 26 years of working within further and higher education. He was Head of Computing and IT at Gloucestershire College, managing its successful Centre of Vocational Excellence and trailblazing programmes in Interactive Games Design and Forensic Computing at Level 3. He has taught at most academic levels (including pre-16 and post-16), developed and lead written BTEC IT programmes at level 2 and 3 and has led the development of a new nationally recognised Technical qualification in Cyber Security. Currently he is taking a break from academia by pursuing a career in commercial web development after contributing software for an award winning EDF project.

Alan Jarvis

Alan started his career working in the IT industry and then spent 20 years working as a lecturer in Further Education. Alan has contributed to a number of text books and recently has been working with Pearson on qualification development, preparing teaching support materials and training centres in preparation for new versions of the BTEC qualifications.

Richard McGill

Richard is currently a level 3 BTEC course team leader at SGS Filton College with over 25 years experience in delivering and leading BTEC IT and Computing qualifications at this centre. Previously he was a Senior Analyst at Lloyds Bank. He has worked for more than two decades for Edexcel and Pearson with current roles including Senior SV Team Leader, OSCA writer and Lead Examiner. In his spare time, Richard rides a motorcycle and plays bass guitar with a large band in Bristol.

Jenny Phillips

Jenny is an educational consultant advising colleges, training providers, employers and schools on high quality education and training. Jenny fulfilled her ambition to train as a lecturer, whilst working in industry 20 years ago, running her own business and that of others. Jenny's consultancy work involves national and internal travel and varied tasks. She has written many books and articles, multi-media resources and educational film scripts. Her specialist subjects are computing, IT and all aspects of business, having gained 40 related qualifications including several undergraduate and postgraduate degrees in education, research, business and computer science. She is currently working on her doctoral thesis.

Principles of Computer Science 1

Getting to know your unit

Assessment
You will be assessed by a written examination.

The key to computer science is solving problems. In order to solve problems effectively it is necessary to understand them fully, pulling them apart until the required data types, intricate algorithms and programming language syntax start to coalesce into an efficient program that is fit for purpose.

The aim of this unit is to encourage you to transform yourself from a computer program user to a computer program developer, capable of designing and programming smart solutions to new problems by drawing across your full study programme for the skills and knowledge required.

You will learn about computational thinking skills, the common components of different types of programming languages and the tools used to design and develop high-quality applications. In doing so, you will travel through the complete software development life cycle, honing your analytical and problem-solving skills so that you can gain employment as a junior software developer or progress to higher education and specialise further as a computing professional.

How you will be assessed

This unit is assessed through a written examination set and marked by Pearson.

The examination is two hours in length. During the supervised assessment period, learners will be assessed on their ability to apply their computational thinking skills to solve problems.

Grade descriptors

To achieve a grade, you will be expected to demonstrate these attributes across the essential content of the unit. The principle of best fit will apply when awarding grades. The maximum number of marks for this unit is 90.

To pass this unit:

- You will be able to use problem-solving skills to develop a solution to given problems in context.
- You will be able to use standard programming constructs to demonstrate an understanding of how data is handled in a computer program.
- You will be able to construct, propose, develop and explain solutions to a problem and demonstrate an understanding of data validation and error checking.

To gain a level 3 distinction:

- You will demonstrate that you can analyse and interpret problems and develop a detailed and complex solution in response.
- You will demonstrate an in-depth understanding of programming constructs and a thorough understanding of how data is handled in a computer program.

Assessment outcomes

AO1 Demonstrate knowledge and understanding of computing facts, terms, standards, concepts and processes

Command words: complete, draw, give, identify, name, state

Marks: range from 1 to 5 marks

AO2 Apply knowledge and understanding to communicate understanding of computing facts, terms, standards, concepts and processes

Command words: calculate, complete, demonstrate, describe, draw, explain, produce

Marks: range from 1 to 5 marks

AO3 Select and use computing technologies and procedures to explore outcomes and find solutions to problems in context

Command words: calculate, demonstrate, develop, explain, produce

Marks: range from 1 to 6 marks

AO4 Analyse data and information related to computer science in order to predict outcomes and present solutions

Command words: analyse, demonstrate, discuss, produce, write

Marks: range from 6 to 12 marks

AO5 Evaluate technologies, procedures, outcomes and solutions to make reasoned judgements and make decisions

Command words: evaluate, produce, write

Marks: range from 6 to 12 marks

Getting started

Programming is chiefly concerned with problem solving.
Write down a list of the tasks that you think a programmer carries out and the questions they need to answer when designing and building a computer program. When you have completed this unit, discuss with your peers, comparing and contrasting your ideas.

A Computational thinking

Many problems are solved by experienced programmers before they even touch a computer keyboard. You should be able to use computational thinking skills to analyse a problem, identify patterns and break complex tasks down into more manageable chunks so that they become easier to tackle.

Successful computer programming relies on exercising your computational thinking skills. It is these skills which help you to investigate a problem, analyse it methodically and identify potential solutions that you can further develop into working software applications.

Computational thinking skills can be divided into four separate, but interlocking, steps, as shown in Figure 1.1.

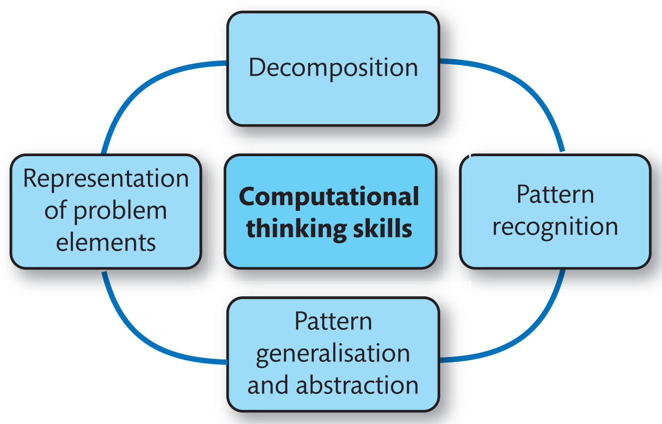


Figure 1.1: Computational thinking skills

You will examine each step in turn in the next section.

Decomposition

Decomposition is the process of breaking complex ideas down into smaller, more manageable parts. Sometimes this process may be called factoring. Generally, problems which are not decomposed prove to be more difficult to solve. The process of breaking a larger problem down into a number of smaller problems often improves the chances of success. This is chiefly because it allows you to focus on just one thing at a time, permitting its details to be examined more closely.

Everyone uses decomposition every day, often without realising. For example, the process of making a family meal involves:

1 choosing an appropriate recipe to follow
2 calculating the correct quantity of ingredients for the recipe and the family size
3 collecting appropriate ingredients
4 preparing the ingredients
5 cooking the ingredients in the right order, using the correct methods and for the correct duration
6 putting the meal on plates, ready to be eaten.

In this way, a single problem (making a family meal) can be decomposed into at least six ordered sub-tasks, each of which could be further decomposed, if necessary, until the steps required to solve each task are relatively straightforward to understand. In programming, decomposition involves the following four stages.

Identifying and describing problems and processes

At this stage, you will list problems and processes concisely, using language that matches the problem's source. For example, if you are dealing with a financial problem, you should accurately use terms from the financial sector. This means that you need to be familiar with the technical language used in the business sector relevant to the problem.

Breaking down problems and processes into distinct steps

At this stage, you will decompose complex problems and processes into separate steps which, when taken together, can be reassembled correctly. There is no specific limit to

the number of steps included or of levels to which you may decompose. You will simply continue to decompose each step until you reach an acceptable level of understanding. For example, the problem of calculating someone's net pay (salary after tax is deducted) is decomposed into several steps in Figure 1.2.

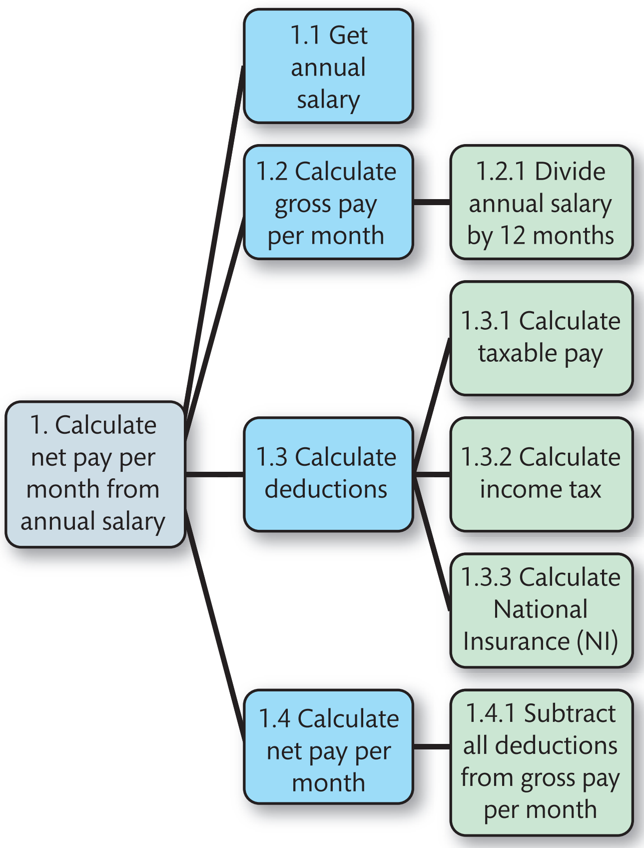


Figure 1.2: Decomposing a single process into progressively smaller steps

Describing problems and processes as a set of structured steps

At this stage, you will document the problems and processes that you have decomposed as a set of structured steps. This should be straightforward enough to be followed by you or by others.

Communicating the key features of problems and processes to others as relevant

At this stage, you will discuss the problems and processes with others. This may include other programmers or the client. Once you have decomposed a complex problem, it is possible to start analysing the steps involved to see if there are any repeating patterns.

Reflect

As a programmer, your communication skills must be flexible. You will need to adjust your delivery to meet the needs of different audiences. For example, programmers will understand technical 'jargon' while a client may not. On the other hand, the client may appreciate the use of business sector-specific language, but programmers may not.

You must be able to communicate a problem to others, because having a really clear understanding of the problem is essential to your eventual success in solving it. Albert Einstein is often quoted as having said, 'If you can't explain it to a six year old, you don't understand it yourself.'

Pattern recognition

Pattern recognition is the ability to see recurring attributes within the same problems and between different problems. For example, a new problem may have features that are similar to problems that have been previously encountered and solved. Recognising these repeating patterns can make problem solving much easier, as it can provide a good place to start.

Pattern recognition is a process based on five key steps.

- Identifying common elements or features in problems or systems. This involves:
  - examining problems or systems
  - listing elements or features that exist in each
  - highlighting those which exist in multiple places
  - recognising these as patterns.
- Identifying and interpreting common differences between processes or problems. This involves:
  - examining problems and processes
  - listing elements or features that exist in each
  - highlighting those that are unique to each
  - recognising these as differences.
- Identifying individual elements within problems. This involves:
  - examining problems to identify the inputs, processes (including selections and iterations) and outputs that are present.
- Describing patterns that have been identified.
- Making predictions based on identified patterns:
  - for each identified pattern, determine how it could be used in the future or how it may appear in similar situations.

Pattern generalisation and abstraction

In computing, abstraction is a concept whereby systems are split into different layers, with each layer hiding the complexity of the layer existing beneath it. This allows a programmer to use a feature without having to know exactly how it works: the irrelevant and intricate mechanics are simply 'abstracted' away or removed.

Pattern generalisation happens when relationships between patterns can be identified and simple conclusions can be drawn. For example, patterns can be identified even when, at first, it does not look like there are many similarities, as shown in the photos.

▶ Repeating design patterns (if you look closely...)

In each of the forms of transport shown in Figure 1.3, you can see similar elements repeated in the design patterns. Each vehicle includes four wheels, two axles, a chassis, a steering mechanism and so on. In order to solve a problem effectively, you must look beyond the obvious (in this case, the physical differences between the three forms of transport). Instead, try to identify the elements that are common and the relationships between these elements.

There are two parts to this phase of computational thinking.

First, **identify the information required to solve an identified problem**. You can achieve this by:

- knowing what information you need
- knowing your reasons for needing this information (the 'rationale')
- knowing the format in which this information needs to be provided
- knowing how soon this information is required to prevent the solution from being delayed.

Second, **filter out the information *not* required to solve an identified problem**. You can achieve this by:

- knowing what information is not needed, as this will be a distraction
- knowing (and justifying) why you have excluded this information.

Representing parts of a problem or system in general terms

To do this, you need to identify the following.

- **Variables** – these are the values in a problem or system that may change, which are typically input by the user or as the result of a required calculation.
- **Constants** – these are the values in a problem or system that do not change often or that remain fixed for a reasonable period of time (e.g. the base rate of income tax being 20%).
- **Key processes** – these are the processes that are absolutely critical to understanding a problem or how a system works.
- **Repeated processes** – these are processes that occur multiple times within a problem.
- **Inputs** – these are the values entered into the system, including the units used and, potentially, any valid values or ranges (e.g. where gender is 'M' for male or 'F' for female, or where a house price has to be between £20,000 and £2,000,000).
- **Outputs** – this is information presented to the user in a required format, which is generally specified by the client as part of their requirements.

Discussion

In small groups or pairs, think about an everyday process such as calculating the cost of decorating a room with fresh paint. Try to determine the variables, constants, key processes, inputs and outputs that are involved.

Algorithm design

An algorithm is simply a set of instructions that is followed to solve a problem or perform a particular stage of processing in the overall solution, for example to validate user inputs. Programs may be made from a collection of many different algorithms.

A core part of the algorithm design is the description of the main processes, the required inputs, the desired outputs and any data storage that is required.

Link

Algorithms used to design mobile apps are discussed in *Unit 17: Mobile Apps Development*.

Worked Example: Creating an algorithm

You are now going to explore a simple algorithm that can be used to convert an amount of UK pounds into American dollars (D) or euros (E), depending on the user's choice.

Step 1: Understand the problem

- *Make sure that you fully understand the problem you have been asked to solve.*

You need to convert an amount of pounds into dollars or euros (D or E), depending on the user's choice.

Step 2: Identify the inputs

- *What values will be input by the user?*
- *What type of data will this be?*

An amount in pounds.

A choice of converting to D or E.

Step 3: Identify the processes

- *What kind of calculations are involved?*
- *Are there any special values we need to know, for example conversion rates?*

You need the conversion rates from pounds to D and from pounds to E.

You also need to validate the currency choice to D or E.

Step 4: Identify the data storage

- *This would mean thinking about storing data that would be needed next time the solution is used.*
- *If this is needed, what type of data do you need to store?*

Data does not need to be stored.

Step 5: Identify the outputs

- *What information must be output?*
- *In what format must the output be shown?*

You need to output the converted sums in a decimal format using the correct currency symbols ($ or €).

Step 6: Collect your notes

- *Use a quad diagram to represent the inputs, processes, data storage and outputs required.*

A breakdown of this problem into these four separate areas is shown in Figure 1.3.

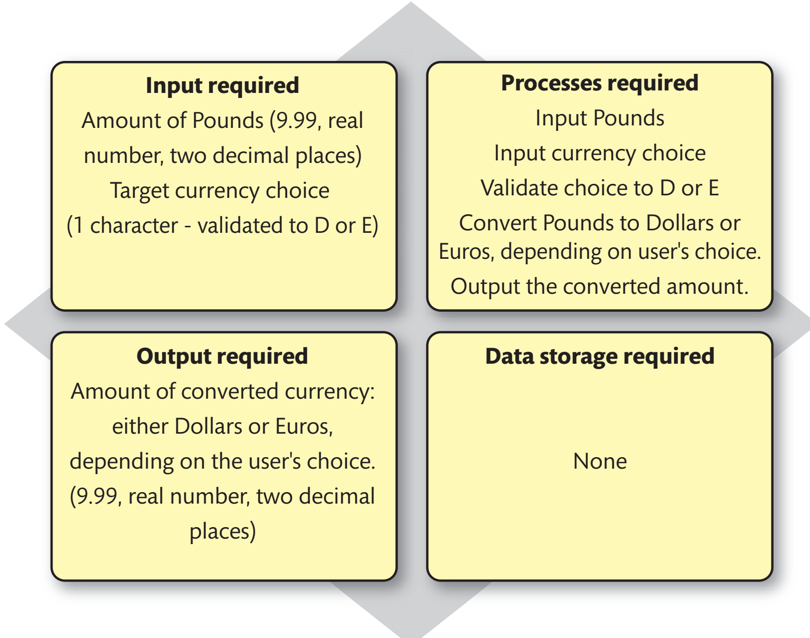


Figure 1.3: A quad diagram showing inputs, processes, outputs and data storage

The quad diagram is a simple way of starting to organise your thoughts while exploring a problem and starting to assemble a potential solution.

The processes you have identified will need to be developed into more detailed algorithms before you can form a potential solution.

In this example, we have listed all the inputs and outputs and noted their formats (particularly the quantity of decimal places to use) and any validation that is required. When specifying formats it is typical to use 9 for a numeric digit, A for an alphabetic character and X for any character.

PAUSE POINT Can you explain what the assessment outcome was about? What elements did you find easiest?

Hint Close the book and make a list of the features of computational thinking skills and the steps you might take to start building an algorithm.

Extend Are there any other inputs, outputs or processes that you think have been missed?

B Standard methods and techniques used to develop algorithms

Because they can be complicated to communicate verbally, algorithms are best represented using a number of different design tools. This section looks at two of the more common tools that can be used – structured English (pseudocode) and flowcharts.

Structured English (pseudocode)

Pseudocode is an informal English-like outline of the algorithm which can be converted to the target programming language.

Pseudocode should avoid commands or syntax that are found only in a particular programming language. Instead, standardised terms from English are used to specify particular types of actions. A categorised list of these is shown in Table 1.1. You may be introduced to others in your studies.

Table 1.1: Categorised pseudocode terms

Operations	Decisions	Repetition
BEGIN	IF	FOR
END	THEN	REPEAT UNTIL
INPUT	ELSE	WHILE
OUTPUT	ELSEIF (ELIF)	WHILE NOT
PRINT	WHEN	
READ		
WRITE		

Figure 1.4 shows pseudocode being used to describe a simple algorithm that validates a user's age between 18 and 60 (inclusive).

```
BEGIN
  REPEAT
    min = 18
    max = 60
    OUTPUT "What is your age?"
    INPUT age
    IF age < min OR age > max THEN
      OUTPUT "Error – age must be between 18 and 60"
    ELSE
      OUTPUT "Age is accepted"
    ENDIF
  UNTIL age >= min AND age <= max
END
```

Figure 1.4: Pseudocode which validates a user's age

You should note the use of **indentation** to highlight the hierarchy of the algorithm's logic. When indentation is used correctly, it should be possible to see which actions are performed in the true or false branches of each decision or which actions are being repeated by a loop.

Key term

Indentation – optional styling applied to pseudocode and the source code of certain programming languages to help convey the code's underlying structure to a reader. Indentation is created through the use of whitespace and the vertical alignment of related code elements. Different indent styles exist and are usually specified as part of an organisation's house rules.

Link

For more on how to produce pseudocode, including sequence, structure, operations, decisions and repetition, see Writing pseudocode, in Structured English (pseudocode) in *Unit 4: Software Design and Development Project*.

Interpreting pseudocode

To succeed in this unit, you must be able to not only produce pseudocode to solve set problems but also be able to correctly interpret existing pseudocode statements so that you can describe the tasks or processes they are performing.

Apply processes to calculate outcomes

Interpreting existing pseudocode involves applying the processes that are shown to determine potential outputs and actions, which will help you to understand the true purpose of the pseudocode presented. You should also be able to evaluate the structure and logic of the pseudocode provided against the original requirements: that is, does the solution meet the target user's needs?

Evaluate the structure and logic of given code against given requirements

As part of any development, you should try to create solutions that demonstrate effectiveness and efficiency of code and identify and fix any errors that may exist within the code.

Suggest improvements to logical structures and processes

Finally, you should be able to suggest improvements to logical structures and processes. Try to interpret the line-numbered pseudocode shown in Figure 1.5 and decide what it is trying to achieve.

```
1   BEGIN
2     INPUT value
3     total = 1
4     counter = 1
5     REPEAT
6       total = total * counter
7       Increment counter
8     UNTIL counter > value
9     PRINT total
10  END
```

Figure 1.5: Sample pseudocode

So, what does this pseudocode algorithm actually do?

If you examine the code line by line, you should be able to work out what it is attempting to do.

Line 1 – indicates the start of the pseudocode.

Line 2 – the user inputs a value, storing it in a variable called 'value'.

Line 3 – creates another variable called 'total', initialising it to 1.

Line 4 – creates a variable called 'counter', initialising it to 1.

Line 5 – starts a looping process.

Line 6 – adds 'counter' times 'total' to the existing 'total'.

Line 7 – increments 'counter' (increment means to increase by 1).

Line 8 – a loop condition checks the value of 'counter'; if 'counter' is greater than the value input in line 2, the loop will stop, otherwise, it returns to line 5 and performs the actions again.

Line 9 – prints the 'total' variable on screen.

Line 10 – indicates the end of the pseudocode.

Link

See Variables, in Pattern generalisation and abstraction, for a definition of variables. For more about loops, see Loops, in Control structures and, for a definition of operators, see Operators, in Arithmetic operations.

If you input 4 for 'value' and trace the values of 'counter' and 'total' line by line, you will see the following results (Table 1.2).

Table 1.2: Tracking pseudocode loops and variables

Value	4			
Counter	1	2	3	4
Total	1	2	6	**24**

Do you recognise this pattern? This is four factorial (denoted by 4!) which is 4×3×2×1 = 24. This pseudocode algorithm generates factorials from an inputted value. Algorithms can get much more complicated than this, of course, but identifying this is a good start.

Improve the effectiveness and efficiency of code

As part of any development, you should try to create solutions that demonstrate effectiveness and efficiency of code and identify and fix any errors that may exist within the code.

Link

See also Improve the effectiveness and efficiency of code, in Developing pseudocode, in *Unit 4: Software Design and Development Project*.

Identify and fix errors

Sometimes, errors will occur in the pseudocode and these may be difficult to find. For example, examine the following code (shown in Figure 1.6) and try to identify the error.

```
1   BEGIN
2     INPUT value
3     total = 1
4     counter = 1
5     REPEAT
6       total = total * counter
7     UNTIL counter > value
8     Increment counter
9     PRINT total
10  END
```

Figure 1.6: Sample pseudocode with an error

If you examine the code carefully, particularly between lines 5 and 8, you will see the problem.

In this version of the pseudocode the REPEAT...UNTIL loop stops when the 'counter' becomes larger than the 'value' entered by the user. However, if you look closely, you will see that the line of code which increments the 'counter' variable is not performed until line 8 – it is outside the loop, which cannot be correct. Doing this means that you will never make the 'counter' variable increase inside the loop. As a consequence, the loop condition will never become true and it will never end – you have created an 'infinite' loop.

To fix this, you would need to move line 8 back inside the REPEAT...UNTIL loop. Such errors can be very subtle, but you will be expected to be able to identify them and fix them. This process is called debugging.

Link

See also Identifying and fixing errors within pseudocode, in Developing pseudocode, in *Unit 4: Software Design and Development Project*.

Flowcharts using standard symbols

A flowchart is a graphical representation of the algorithm, showing its actions and logic through a set of standardised symbols which are standardised by the **British Computer Society (BCS)**.

Key term

British Computer Society (BCS) – the Chartered Institute for IT, which collaborates with the Government and industry to set and promote common standards within IT.

Link

To learn more about the flowcharts and the common BCS flowchart symbols, see Flowcharts and the use of standard symbol conventions, especially Table 4.1, in *Unit 4: Software Design and Development Project*.

Worked Example: Creating a flowchart

You are now going to use these symbols to build a flowchart that will represent the age validation algorithm as shown in figure 1.4.

Step 1

Think about the operations you need to perform, for example inputs, processes, outputs.

Step 2

Place them into a possible sequence.

Step 3

Are there any decisions to be made?
Are any actions dependent on these decisions?

Step 4

Are there any loops/repetitions required?
What conditions control the loops?

Step 5

Select the correct symbols needed and start to match these to the content already decided.

Step 6

Draw the flowchart, remembering to make sure that all flowchart symbols are correctly connected using arrows which demonstrate the correct logic flow when the flowchart is read.

Figure 1.7 shows the age validation algorithm using flowchart symbols.

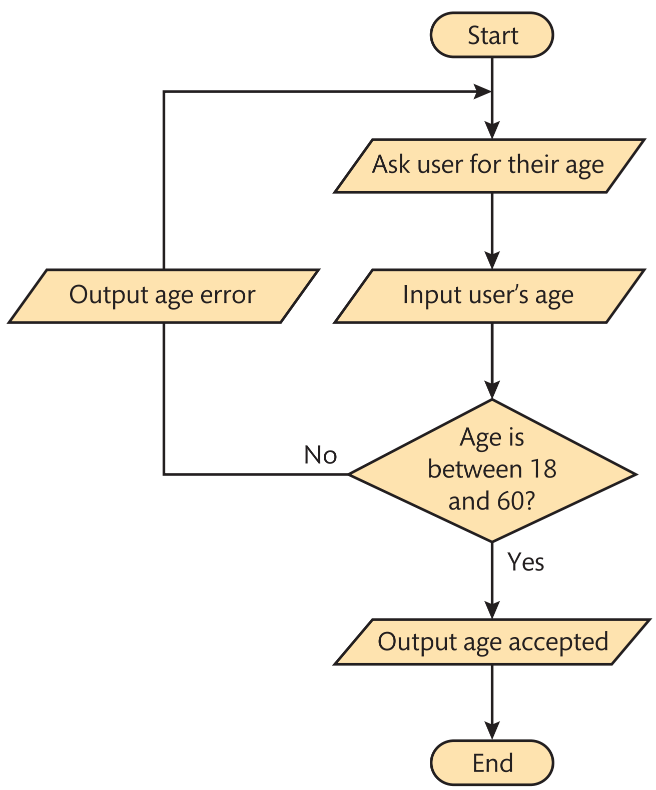


Figure 1.7: Flowchart representation of a simple algorithm

PAUSE POINT Can you explain what the assessment outcome was about? What elements did you find easiest?

Hint Close the book and make a list of common pseudocode operation, decision and action words.

Extend You can create algorithms using any standard design method or technique. Which method do you think is easier to use – flowcharts or pseudocode? Why do you think this is?

Assessment practice 1.1

A01 A02 A03

Read through the following scenario and using computational thinking skills (decomposition, pattern recognition, pattern generalisation and abstraction) solve the problem using standard methods and techniques such as pseudocode and flowcharts.

An online payment system, myBill, allows users to effortlessly pay for a variety of website purchases on the internet. Each myBill customer can link their account to a bank account, debit or credit card. Amounts are debited from the preferred funding source to create a myBill balance. Usually, myBill only takes what is needed to pay a transaction so the myBill balance remains at zero. However, sometimes a website will refund part of a transaction leaving the myBill balance in credit. When this occurs any subsequent purchase has to use the myBill credit first and then subtract the difference from the preferred funding source.

A simulation of this process is required for demonstration purposes where users can select their preferred funding stream, process a transaction (payment or refund) and keep an accurate running myBill balance. Each funding source set up should have its own balance and if a myBill transaction occurs when the preferred funding stream is unable to provide sufficient money, then the transaction should be politely refused.

Plan

- Do I fully understand the nature of the problem?
- Am I clear about what I am being asked to do?
- Do I know how to apply computational thinking skills to solve the problem?
- Am I able to use standard methods and techniques, such as pseudocode and flowcharts, to solve the problem?

Do

- I know what it is that I am doing and what I want to achieve.
- I can use a variety of computational thinking skills to analyse the problem and break it down into much more manageable chunks.
- I will start from the beginning of the scenario and work through to the end.

Review

- I can explain the computational skills I have used.
- I can explain how pseudocode and flowchart solutions both help to represent the algorithms I have created.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

C Programming paradigms

In this section, you will learn about standard structures and conventions (programming paradigms) used to build and develop accurate, efficient and effective computer code to fulfil identified criteria and solve problems.

Handling data with a program

Data is typically represented in a program using an **identifier**. Many types of identifier are used in programs, but the most common types are constants (an identifier representing a value that will not change while the program is running) and variables (an identifier representing a value that may change while the program is running).

Key term

Identifier – a programmer-friendly name which represents a value stored in the computer's RAM. Before the use of identifiers, a programmer would need to know the actual memory address of a value in order to access it.

For example, when a person logs into a program, their name and password would typically be stored as variables, as these details would be different for each different user. However, a program calculating the price of a new television in a shop's electronic till would need to ensure that the current rate of value added tax (VAT) is added to the price. This rate would not change for every sale, so it could be set as a constant and only be modified if it changed in the Chancellor of the Exchequer's budget.

Defining and declaring constants and variables

In order to create (or declare) an identifier, whether it is a constant or a variable, you usually provide the programming language with two things: a name and a data type.

Almost all programming languages support the concept of data types. A data type is used to define what kind of value a variable or constant can store, what operations can be performed upon it and its behaviour within the program. Most programming languages offer many different data types for the programmer to use.

You should always choose a sensible and meaningful name. For example, a good variable name for storing the user's name would be 'username'. Some names cannot be used because they are reserved by the language for a particular use, typically because they are command words. These 'reserved' words vary between different programming languages, so be careful.

Link

A list of reserved words for JavaScript is shown in *Unit 17: Mobile Apps Development*.

To learn about the most common data types used in computer programs to define and declare constants and variables, see Defining and declaring constants and variables, in *Unit 4: Software Design and Development Project*.

Managing variables

Once you have established the variables for a program, they will need to be managed by determining what type of variable they are and by applying naming conventions.

Local and global variables

Most programming languages contain the concepts of **local and global variables**.

Key terms

Local variables – these are limited to being used in the block of code in which they are declared, for example within a particular function.

Global variables – these can be used anywhere in your program code.

Link

To learn more about local and global variables, see Local and global variables, in Managing variables, in *Unit 4: Software Design and Development Project*.

Naming conventions

Apart from selecting meaningful identifiers, you will find that many different naming conventions exist for creating identifiers, although these will vary depending on the programming language you are using.

Two widely popular techniques that are used by industry professionals are **snake_case** and **camelCase**. Table 1.3 demonstrates some variables named using these two naming conventions.

Table 1.3: snake_case and camelCase

Identifier needed for:	snake_case	camelCase
user's password	user_password	userPassword
gross pay	gross_pay	grossPay
customer's postage charge	customer_postage_charge	CustomerPostageCharge

As you can see, snake case uses an underline to link multiple words together. Camel case prefers to capitalise the first letter of each word and close any spaces; making the first letter of the first word uppercase is optional.

Research

Many software publishers recommend standards for naming identifiers. Microsoft® freely provides online documentation containing advice and guidance on these technical issues with the aim of encouraging good programming practices. Visit the following online resource to find out about their recommended identifier naming conventions:

https://msdn.microsoft.com/en-us/library/ms229045(v=vs.110).aspx

Link

For more on the topic of naming variables, see Naming conventions, in Managing variables, in *Unit 4: Software Design and Development Project*.

Arithmetic operations

The majority of algorithmic solutions that you design will involve the use of arithmetic operations. These can be categorised as mathematical **operators**, relational operators, Boolean operators or date/time.

Key term

Operators – these are special symbols (or multiple symbols) which tell the program to perform a specific arithmetic, relational or logical operation on its data. Operators have to be used in a specific order of precedence (this describes the order of execution). You may be familiar with this concept from using BIDMAS or BODMAS (Brackets, Indices/pOwers, Divide and Multiply, Add and Subtract) in mathematics.

Link

To learn more about these arithmetic operators, see Arithmetic operations, in *Unit 4: Software Design and Development Project*.

Mathematical operators

The four core arithmetic operators will be familiar from your mathematics studies: add, subtract, divide and multiply. In computer science, symbols that are different from those you will be familiar with are often used, so be careful.

Link

See Table 4.7 and the section Mathematical operators, in Arithmetic operations, in *Unit 4: Software Design and Development Project* for the core mathematical operators.

In addition to the four core mathematical operators, there are also **modulus division** and **integer division**, the symbols for which are shown in Table 1.4.

Table 1.4: Modulus and integer division operators

Operators	Traditional symbol	Computer science operator
Modulus division	Mod	% MOD Modulo Rem
Integer division	/	DIV

- **Modulus division** simply divides two numbers and returns the whole number and remainder. This is probably the type of division you performed before you understood fractions and decimals.
  For example, you are probably familiar with the 24-hour clock.
  Well, you are likely to perform a modulus division of 12 to convert 15:00 to 3pm.
  For example 15 mod 12 = 1 remainder 3
- **Integer division** is simply when one integer number (whole number) is divided by another and you are only concerned with the integer part of the result. In other words, the remainder (decimal) is discarded, e.g.
  Normally 10 / 4 = 2.5
  But 10 DIV 4 = 2

Relational operators

These types of operators define the relationship between two different values. Generally speaking, there are six operations that you need to be familiar with.

Link

To learn about the six relational operators that you need to know about, see Relational operators, in Arithmetic operations, in *Unit 4: Software Design and Development Project*.

Boolean operators

Boolean or logical operators are used to combine expressions together. They give you the ability to test multiple conditions.

Link

To learn more about Boolean operators, see Boolean operators, in Arithmetic operations, in *Unit 4: Software Design and Development Project*.

Date/time

Many programming languages that support date and time data types use a variety of functions and operators to perform basic arithmetic on them. For this reason, it is difficult to identify particular operators for performing calculations with dates and times. However, Figure 1.8 demonstrates the use of simple date/time arithmetic in Python® through the use of the common + and – operators.

```
# Date/Time arithmetic in Python
import datetime

#grab date and time
today = datetime.date.today()
now = datetime.datetime.now()

#specify differences
daydiff1 = datetime.timedelta(days=14)
daydiff2 = datetime.timedelta(weeks=3)
timediff1 = datetime.timedelta(hours=4)

#display calculated differences
print "Two weeks from now is:", today + daydiff1
print "Three weeks ago was:", today - daydiff2
print "In four hours the time is:", now + timediff1
```

Figure 1.8: Python's® use of + and – operators to perform date/time calculations

Link

To learn more about how to apply date and time data types in C++, see Date and time, in Arithmetic operations, in *Unit 4: Software Design and Development Project.*

Built-in functions

Almost all programming languages have built-in library functions which programmers can use when solving complex problems. These allow the programmer to perform common but crucial tasks on data, such as formatting its appearance, finding the length of a string or the square of a number. You can also download and install third-party functions from reputable websites to expand programming languages.

Built-in functions are grouped by category: arithmetic functions, string handling functions or general functions.

Arithmetic functions

Arithmetic functions perform mathematical operations.

- Random

Random numbers are important for generating test data, particularly for statistics and probability applications and introducing the element of unpredictability into a program, e.g. for simulation or games of chance such as dice, roulette or card-based games such as pontoon or poker.

Most programming languages support the concept of pseudorandom numbers. These numbers are often generated (through hardware or software algorithms) via a process of manipulating seemingly random events (real-time clock readings, input/output activity etc) to generate a pool of random values.

It is quite common for the pseudorandom number generator (PRNG) to be given an initial seed value that acts as a starting point for the creation of the pseudorandom number pool. If multiple starting points are chosen too quickly in succession, e.g. inside a short loop, it is possible for a PRNG to generate the same sequence of random numbers, which essentially defeats the objective, so caution is advised.

Most programming languages have built-in functions which return a random number within a requested range. Figure 1.9 shows a typical C# method for generating and displaying pseudorandom numbers.

```
using System.IO;
using System;

class Program
{
    static void Main()
    {
        //create array for months of the year
        string[] months = new string[12]
        {"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"};

        //initialise prng with a time-dependent default seed value
        Random rnd = new Random();

        // will create a number between 1 and 12
        int month = rnd.Next(1, 13);

        //output the random month name
        Console.WriteLine("Month is {0}", months[month-1]);
    }
}
```

Figure 1.9: Selecting a random month using a built-in PRNG in C#

Range

Although many languages have similar functions, Python's® Range function is an example of a good utility function. Despite being able to arithmetically generate a list of numbers, its uses can be quite general-purpose, especially when used in conjunction with FOR loops.

Range is a flexible function, accepting a variety of different arguments as shown in Figure 1.10.

```
# range function examples

# One argument, generates numbers from 0 to 5
for i in range(6):
 print(i)

# Two arguments, generates numbers from 2 to 7
for i in range(2, 8):
 print(i)

# Three arguments, generates numbers from 1 to 10 in steps of 2
for i in range(1, 10, 2):
 print(i)

# Going backwards, counting down from 10 to 0 in steps of 2
for i in range(10, 0, -2):
 print(i)
```

Figure 1.10: Python® range function

The next example as seen in Figure 1.11 uses the range function to generate the counter values to traverse the array's 7 elements using a FOR loop.

```
# More complex example of range function in Python

days = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']
size = len(days)

for counter in range(0, size):
    print(days[counter])
```

Figure 1.11: Use of range function to control a FOR loop

Research

Investigate other programming languages to see if functions are available with similar functionality to Python's® Range function.

Round

An important objective when dealing with any problem involving money or quantities is the ability to round decimal numbers to a desired number of places or simply round to an integer (whole number).

As you might expect, most programming languages contain built-in functions to perform a simple rounding function although you should take care with numbers such as 1.5 as some rounding functions will round up and others may unexpectedly round down.

Figure 1.12 shows a simple example of using a C# built-in function to round a decimal number to a preferred number of decimal places.

```
using System.IO;
using System;

class Program
{
    static void Main()
    {
        //original floating point values
        float originalNumber1 = 1.23f;
        float originalNumber2 = 3.15f;
        float originalNumber3 = 1.9f;

        //rounded numbers - parameters are original number, dec places
        float roundedNumber1 = (float)Math.Round(originalNumber1, 1);
        float roundedNumber2 = (float)Math.Round(originalNumber2, 1);
        float roundedNumber3 = (float)Math.Round(originalNumber3, 0);

        //output the rounded numbers
        Console.WriteLine("roundNumber1 is 1.2, roundedNumber1);
        Console.WriteLine("roundNumber2 is 3.2, roundedNumber2);
        Console.WriteLine("roundNumber3 is 2, roundedNumber3);
    }
}
```

Figure 1.12 Using the round method of the Math class as a built-in function

Additional functions may be available which always round down ('floor') or always round up ('ceiling') as shown in Figure 1.13.

```
using System.IO;
using System;

class Program
{
    static void Main()
    {
        //original floating point value
        float originalNumber = 2.45f;

        //rounded numbers - parameter is the original number
        float roundedNumber1 = (float)Math.Ceiling(originalNumber);
        float roundedNumber2 = (float)Math.Floor(originalNumber);

        Console.WriteLine("roundNumber1 is 3, roundedNumber1);
        Console.WriteLine("roundNumber2 is 2, roundedNumber2);
    }
}
```

Figure 1.13: Using floor and ceiling

In this example, ceiling would give 3.0 and floor would give 2.0; the variables are still floating point numbers. However, you must remember that a negative value such as –2.8 would 'floor' to –3 and 'ceiling' towards –2.

- Truncation

In classical programming, truncating a number should simply 'chop off' the decimal part of a floating point number leaving the integer part behind.

Link

To learn more about common arithmetic functions, see Arithmetic functions, in Built-in functions, in *Unit 4: Software Design and Development Project*.

String handling functions

Although strings are simply a collection of characters, they may still require processing as part of any potential solution. This task is commonly known by programmers as string handling. Programmers often have the need to perform certain operations on strings and most high-level languages have built-in library functions to help.

Common tasks are shown below, with C# practical code examples showing their implementation.

- Find the length of a string in characters

```
string month = "January";
int length;

length = month.Length;
Console.WriteLine("{0} is {1} characters long", month, length);
```

In this example, the **length** property of the 'month' **string object** is used to access the string's number of characters.

- Examining single characters

```
string month = "January";
char singleLetter;
int whichLetter = 3;

singleLetter = month[whichLetter];
Console.WriteLine("Character {0} of {1} is {2}", whichLetter, month, singleLetter);
```

In this example the output will be:

```
Character 3 of January is u
```

This is because 'u' is the character in position 3; the string starts with a 'J' in position 0.

- Extract a sub-string (part of a string)

```
string month = "January";
string subString;
int startPos = 3;
int howMany = 2;

subString = month.Substring(startPos, howMany);
Console.WriteLine("Substring is {0}", subString);
```

In this example the output will be:

```
Substring is ua
```

This is output because we start extracting at position 3 ('u'), and we want to extract the next 2 characters, including the first one ('ua').

- **Concatenating** two strings

```
string month = "January";
string saying = " has 31 days.";
string combined;

combined = month + saying;
Console.WriteLine("Joined string is {0}", combined);
```

In this example the output will be:

```
Joined string is January has 31 days.
```

This is output because we use the '+' operator to join the two strings together. Other techniques are available in C#, but this is perhaps the simplest to use.

- String conversion

It is important to understand that numeric values can be stored as integers (if 'whole') or floating point (if they need a decimal part) or even as a string.

If the number is stored in a string it is generally impossible to perform arithmetic on it. Sometimes this might be ok; a telephone number typically only has digits and could safely be stored in a string as it is unlikely we would want to add or subtract with it.

However, sometimes we may legitimately want to convert from a number (integer or floating point number) to a string or vice versa. Most programming languages do this via built-in functions or data type **casting**.

> **Key terms**
>
> **Concatenation** – is the process of joining things together. In programming, it is used to describe the process of joining shorter strings together to form longer ones, e.g. 'Hello' and 'World!' to form 'Hello World!'.
>
> **Casting** – is where a value is converted from one type of data to another, potentially losing data during the process, e.g. when truncating decimal places when converting a fractional number to an integer. Casting can be implicit (the language automatically handles it for the programmer) or explicit (the programmer has to specify the conversion).

Integer/floating point to string

Sometimes it is necessary to convert a numeric value (integer or floating point) to a string representation. This may happen because we want to format it in a particular way, e.g. as currency with a £ or $ symbol prefixed or simply to store it in a text file.

Figure 1.14 demonstrates how Python® converts integers and floating point numbers to strings.

```
# Converting numbers to strings in Python

# store the numbers
num1 = 1.4
num2 = 99

#convert them using str() function
string1 = str(num1)
string2 = str(num2)

#prove they are strings
print "String1",string1,"is",len(string1),"character(s) long"
print "String2",string2,"is",len(string2),"character(s) long"
```

Figure 1.14: Python® conversion of numbers to strings using built-in functions

String to floating point number

This C# example demonstrates the use of Parse method (function) of the float class.

```
using System.IO;
using System;

class Program
{
    static void Main()
    {
        string mystring = "123.45";
        float mynumber = 0;

        mynumber = float.Parse(mystring);
        Console.WriteLine("My number, 100 times bigger is {0}", mynumber * 100);
    }
}
```

We can prove the number has been successfully converted by performing a simple arithmetic operation on it, in this case multiplying it by 100.

String to integer

This C# example demonstrates the use of the Parse method (function) of the int class.

```
using System.IO;
using System;

class Program
{
    static void Main()
    {
        string mystring = "123";
        int mynumber = 0;

        mynumber = int.Parse(mystring);
        Console.WriteLine("My number, subtract 1 is {0}", mynumber - 1);

    }
}
```

Once again, we prove the number has been successfully converted by performing a simple arithmetic operation on it, this time by simply subtracting 1.

General functions

Most programming languages that execute on the command line interface (CLI) or shell have functions or commands that allow the user to input data and output messages. These are both fairly fundamental objectives for any programmed algorithm and you are certain to encounter them in any programming language.

The following examples focus on functions found in the Python® programming language.

Input

Python's® input function is used to store user input that has been keyed. It does so by displaying a user prompt, waiting for keyboard input and then storing this in a specified variable in RAM once the enter/return key is pressed. Figure 1.15 shows this Python® code and on-screen prompt.

```
# Simple use of input function in Python to get user input

username = input('Enter your name: ')
```

Figure 1.15: Python® input function

Link

To learn more about common string handling functions, see String handling functions, in Built-in functions, in *Unit 4: Software Design and Development Project*.

Link

To learn about the built-in general functions in C++, see General functions, in Built-in functions, in *Unit 4: Software Design and Development Project*.

Tip

This use of the Input function is specific to Python® 3; in earlier versions, the raw_input function should be used instead.

Research

The majority of programming languages need to permit keyboard input and screen output.

Compare the use of the Python® **print function** shown in Figure 1.16 with the C# **Console.WriteLine function** shown in Figure 1.18.

What is C#'s equivalent to the **input function**?

As you can see, the use of the prompt is important because, without it, the user would not know what the program is expecting from them (or that the program has halted for their input). Of course, at the moment, you are not doing anything with the input.

Print

Python® also has the ability to display messages on the CLI/shell using the Print function. The example shown in Figure 1.16 expands the previous example so that the computer responds with the number of characters in your input name by using the Print function.

```
# Simple use of input function in Python to get user input

username = input('Enter your name: ')

length = len(username)

print('Hello,', username, '!', sep='')
print('Your username is', length, 'character(s) long.', sep=' ')
```

Figure 1.16: Use of Python's® print function to display text and variables

You should notice that the 'sep' argument is used to specify the character to use between the different parts of the print string. If it is not specified, the default separation is a single space.

Open, close, read and write

Designing any software solution typically involves dealing with persistence of data. A computer's data storage in its random access memory (RAM) is said to be volatile – it is lost when power is removed, that is, when the computer is switched off.

This means that it is fine to store data in RAM while the program is running but to have data existing between program uses, especially when the power is removed, requires the use of a more permanent (non-volatile) form of data storage, for example magnetic storage, such as a hard disc, or a universal serial bus (USB) flash drive.

The most common form of non-volatile data storage is the data file and these are supported by most programming languages, including Python® and C#.

Basic Python® data files can be explored using just four main built-in functions:

- **Open** – opens a file
- **Read** – reads data from a file into RAM
- **Write** – writes data from RAM to a data file
- **Close** – closes a file.

The example shown in Figure 1.17 demonstrates the creation of a simple data file (in ASCII text file format) and its reading back into the RAM to verify its contents.

```
# Simple example showing the creation and reading of a data file

# Open a new file for writing
myfile = open("demo.txt", "w")

#write data to the file
myfile.write( "This is my test data");

# Close the opened file
myfile.close()

# Re-open the file, this time for reading
myfile = open("demo.txt", "r")

#write data to the file
message = myfile.read();

#print the contents
print ('Data stored in file was:',message,sep=' ');

# Close the opened file again
myfile.close()
```

Figure 1.17: Basic Python® data file handling

Research

Investigate other programming languages to see if built-in functions are available with similar functionality to Python's® Input, Print and Output functions. C++ is covered in *Unit 4: Software Design and Development Project* but you should also research other programming languages.

Validating data

You may have heard of the expression 'garbage in, garbage out' or GIGO, which suggests the rule that the quality of output is directly dependent on having sensible inputs. **Validation** is the process that attempts to prevent nonsensical inputs.

Although the use of checkboxes, buttons and list boxes limit input choices, most programs still require validation to handle the probability of problematic input from the user, particularly when using traditional keyboard input.

Validation check techniques

The term "constraints" is often used to describe limiting factors which can be used to validate user inputs. Although constraints are often available as special functions, e.g. NotNull, NotBlank, etc. in certain programming languages, they can be created using standard validation check techniques, as shown throughout the following section.

In this section, you will look at a variety of different validation check techniques that programmers can use to validate their programs.

It is very important to build validation rules into a solution. They will check whether different inputs are sensible and prevent inaccurate results, **run-time errors** or fatal application crashes.

Range check

A range check assesses whether data entered is within a valid minimum-to-maximum range. For example, if a customer is limited to buying up to 10 of a particular item, then the valid range would be 0 to 10. Input outside this inclusive range would be considered invalid. Typical C# program code to perform this type of validation is shown in Figure 1.18.

Key terms

Validation – a process that checks to see if an input value makes sense and is reasonable before it is processed. Validation does not check to see if the data has been keyed in accurately – this is a separate process called verification.

Run-time error – a problem that occurs while an application is being used. These errors result in the application locking (refusing to accept user input) or crashing (terminating and returning the user to the device's menu or desktop).

Link

To learn about other validation check techniques (data type, range, constants and Boolean), see Validation check techniques, in Validating data, in *Unit 4: Software Design and Development Project*.

```
const int MIN = 0;      //minimum value of the range
const int MAX = 10;     //maximum value of the range

string strNumber;       //string to temporarily store our input
int qtyValue;           //our inputted quantity

//perform loop while quantity is outside range
do
{
    //input quantity
    Console.WriteLine("Enter quantity between {0} and {1}:", MIN, MAX);
    strNumber = Console.ReadLine();
    qtyValue = int.Parse(strNumber);

    //check is quantity outside range
    if (qtyValue < MIN || qtyValue > MAX)
    {
        Console.WriteLine("Sorry, the quantity entered is outside the allowed range");
    }

} while (qtyValue < MIN || qtyValue > MAX);

Console.WriteLine("Your quantity of {0} is valid, thank you.",qtyValue);

//@TODO other things with the quantity...

//wait for keypress to continue
Console.ReadKey();
```

Figure 1.18: Validation routine for a defined range in C#

Of course, an input of 0 would indicate that the customer is not trying to purchase any number of the item. However, it is still a logically valid input and would pass validation.

Ranges do not have to be limited to numerical input. For example, the characters 'A' to 'F' could also represent a valid range.

Length check

A length check assesses how many characters have been entered. Some very well-known inputs have limited length, for example:

- short message service (SMS) texts are limited to 160 characters
- tweets were originally limited to 140 characters (derived from SMS length minus 20 characters for the user's unique address).

Figure 1.19 shows a C# extract which limits character input to a predetermined maximum length set by a constant.

```
const int MAX = 10;     //set maximum number of characters

string message;         //our message to input and process
int messageLength;      //our message's length in characters

//perform loop while string too long!
do
{
    //input string
    Console.WriteLine("Enter message (max {0} characters)", MAX);
    message = Console.ReadLine();

    messageLength = message.Length;

    //check its length
    if (messageLength > MAX)
    {
        Console.WriteLine("Sorry, your message of {0} characters is too long.",messageLength);
    }

} while (messageLength > MAX);
```

Figure 1.19: Validation routine for a typical length check in C#

Presence check

A presence check assesses whether data is present – that is, whether it exists. For example, a user has to input whether they are 'male', 'female' or 'would prefer not to say' when completing a registration input form. The presence check validates that they have not left the entry blank or unselected.

Type check

A type check assesses whether the data entered is of the correct data type, as in Figure 1.20. For example, if the user has to enter their age, this would require the input to be an integer (a whole number). If input of incorrect data types is not prevented by the programmer, it can cause a fatal run-time error.

Age?	A	(data type is character, this is invalid)
Age?	16	(data type is integer, this is valid)
Age?	01/09/16	(data type is date, this is invalid)

Figure 1.20: Validation responses based on a type check

Format check

A format check assesses whether the data entered is in the correct format: for example, whether a string containing a UK postcode follows the format 'PO1 3AX'.

- PO is the area, such as GL (Gloucester) – this has to be capitals, alphabetic, 1 or 2 characters.
- 1 is the district, usually between 1 and 20 per area – this has to be up to 2 numeric digits.
- 3 is the sector, usually covering up to 3000 addresses – this has to be 1 numeric digit.
- AX is the unit, usually covering up to 15 addresses – this has to be capitals, alphabetic, 2 characters.

A format check on a postcode would apply these rules as shown in Figure 1.21.

Postcode?	GL2 4TH	(postcode is valid)
Postcode?	W1A 1AA	(postcode is valid)
Postcode?	GL2 24TH	(postcode is invalid; sector can only be 1 numeric digit)

Figure 1.21: Validation responses based on a format check

Check digit

A check digit is a single character (usually a numeric digit) derived from an algorithm which is performed on a piece of data. The algorithm is designed to only generate this particular digit if the data (e.g. a string of characters) has exactly those characters and they are arranged in that specific order. Any incorrect character or swapping of character positions generates a different check digit value and fails the test. Check digits are used mostly to detect errors in inputted values such as barcodes, bank account numbers and software registration codes.

An ISBN-10 (10-digit International Standard Book Number) uses a check digit. For example, the previous version of this very book had an ISBN-10 of 1846909287. The last digit (7) is treated as the check digit. You can determine if this code is correct using the follow technique.

Step by step: Check digit validation

4 Steps

1 Multiply each digit by smaller and smaller positional weights, as shown in Table 1.5.

Table 1.5: Positional weights used to check an ISBN-10 checksum validation

Digits in ISBN	1	8	4	6	9	0	9	2	8	7
Factor of multiplication	×10	×9	×8	×7	×6	×5	×4	×3	×2	×1
Sub-totals	**10**	**72**	**32**	**42**	**54**	**0**	**36**	**6**	**16**	**7**

2 Sum or add up the sub-totals in all columns:

10 + 72 + 32 + 42 + 54 + 0 + 36 + 6 + 16 + 7 = 275

3 Perform **modulus division** by 11:

275 MOD 11 = 25 remainder 0

4 Is there a remainder of 0? Yes, there is, which means that the ISBN is valid. Any other result means an incorrect ISBN has been entered

Key term

Modulus division – this performs a division operation and returns the remainder rather than working out the decimal or fractional answer.

This technique identifies incorrect or transposed digits in the code. In the example given in the step by step, the last digit (7) is considered to be the check digit.

Spelling

A spelling check assesses whether the words being entered can be found within an electronic dictionary – that is, that they are valid words.

Error handling and reporting

Many modern programming languages have syntax features which are designed to handle run-time errors when they occur. If they did not have these features, the application would crash or lock unresponsively.

A common error handling technique is the use of the 'try...throw...catch' expression that can be found in many languages, including C++, Microsoft® C#, Oracle® JavaScript and PHP. This error handling technique works by 'trying' an operation, 'catching' any possible errors and (optionally) 'throwing' an appropriate exception. This approach prevents the application from failing the operation in an uncontrolled way, as this would cause the application to crash completely.

An example of this error handling technique is shown in Figure 1.23 (in the next section, Post-check actions) by illustrating the dangers of dividing by zero. In the example shown in Figure 1.23 the division operation is placed inside a try block, just in case the user has entered '0' (zero) as their second number. This has been done because dividing a number by zero can generate a serious error on a computer platform and may result in a run-time crash. By using the 'try..catch' block this is avoided by displaying the exception that has been caught, which, in this case, is a "DivideByZero" error.

Post-check actions

Error reporting is an important aspect of software development. It may be directed towards the user, by telling them that they have made a mistake, or towards the developer, so that they can understand where a program has encountered an error and the nature of this error.

Common techniques for error reporting include:

- displaying an on-screen error message and/or error code
- appending the error details to an electronic log file which can be viewed separately
- sending an email to the developer that includes the error details.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace test
{
    class Program
    {
        static void Main(string[] args)
        {
            float num1;        // first number
            float num2;        // second number
            float quotient;    // result of dividing first number by second
            string strNumber; // string to temporarily store our input

            quotient = 0;

            //get 1st number
            Console.WriteLine("Enter 1st number");
            strNumber = Console.ReadLine();
            num1 = float.Parse(strNumber);

            //get 2nd number
            Console.WriteLine("Enter 2nd number");
            strNumber = Console.ReadLine();
            num2 = float.Parse(strNumber);

            //try the division
            try
            {
                quotient = num1 / num2;
                Console.WriteLine("{0} / {1} = {2}", num1, num2, quotient);
            }
            catch (DivideByZeroException e)
            {
                //output the handled exception
                Console.WriteLine("Error exception caught was: {0}", e);
            }

            //wait for key press before finishing
            Console.ReadKey();
        }
    }
}
```

Figure 1.22: Microsoft® C# example of error handling using try and catch blocks

Tip

You should always take care to not display debug information on the user's screen. This is because it may accidentally reveal sensitive data or the inner workings of the program. This is particularly important when developing online applications in order to prevent hacking and potential fraud.

Control structures

The algorithms that control programs are built using a combination of three basic programming building blocks or control structures.

These control structures are sequence, selection and iteration.

1. Sequence – one action after another, none missed, none repeated.
2. Selection – actions chosen based on a supplied condition which is evaluated to true or false.
3. Iteration – actions repeated a number of times until a condition is no longer true.

Link

To learn more about post-check actions, see Post-check actions, in Validating data, in *Unit 4: Software Design and Development Project*.

It is quite common to see iterations referred to simply as loops and for selections to be called branches.

Loops

By their nature, loop control structures allow parts of an algorithm to be repeated. Although infinite loops are designed to repeat forever, most loops have controlling conditions that tell the loop when to stop or keep repeating. Loop conditions are formed in a similar way to those used in branches.

Most programming languages support a number of different loop types but, in general, they can be classified as being either pre- or post-conditioned. Pre-conditioned loops, such as FOR and WHILE, have a controlling condition that is situated before the actions which need repeating. If the controlling condition is not true, the loop will not work at all and these actions are not executed. The pseudocode for each pre-conditioned loop is shown in Figure 1.23.

FOR loop	WHILE loop

`FOR counter = 1 to 10`
 	
`    OUTPUT counter`
 	
`ENDFOR`
	`counter = 1`
	
	`WHILE counter <= 10`
	
	`    OUTPUT counter`
	`    Increment counter`
	
	`ENDWHILE`

Figure 1.23: Pre-conditioned loops

In addition, the FOR loop is generally differentiated from the WHILE loop because it is ideally used when the number of iterations is fixed and known before the loop starts.

For this reason, it is very useful when traversing arrays, linked lists and other data structures. Additionally, most FOR loops can count up and down and usually increase in set steps, for example jumping in twos or fours.

In contrast, post-conditioned loops such as REPEAT...UNTIL have their controlling condition after the actions being repeated. Because this condition is not tested until after the loop's actions have been executed, you can always rely on the loop working at least once.

REPEAT loop

`counter = 1`
 	
`REPEAT`
 	
`    OUTPUT counter`
`    Increment counter`
 	
`UNTIL counter > 10`

Figure 1.24: Post-conditioned REPEAT loop

BREAK is used to terminate a loop that is designed to work a fixed number of times: usually, a FOR loop. An example of using BREAK is shown in Figure 1.49.

This may seem rather silly, but it makes sense in situations such as linear search routines where a FOR loop may be used to freely traverse data items in an array or list but, once a particular value has been found, there is no need to continue. In this instance, the BREAK can be used to prematurely exit the loop in a controlled fashion.

Link

To learn more about these four kinds of loop, see Loops, in Control structures, in *Unit 4: Software Design and Development Project*.

The C# code for selections and iterations is similar to the JavaScript examples in *Unit 7: Mobile Apps Development*.

Branches

Branches allow you to make decisions within an algorithm, which are usually based on whether a specific condition is true or false.

Conditions are built from combinations of identifiers, **literals**, relational operators and logical operators. For example, to order a holiday online, you may have the following condition, as shown in Table 1.6.

> **Key term**
>
> **Literal** – this is a value in an algorithm or program which can be numeric (e.g. 10) or string (e.g. 'hello').

Table 1.6: Branches example

age >= 18 AND validDebitCard = true						
age	**>=**	**18**	**AND**	**validDebitCard**	**=**	**true**
Identifier (variable)	Relational operator	Numeric literal	Logical operator	Identifier (variable)	Relational operator	Boolean literal

If the condition evaluates to true, one set of actions is performed. If the condition evaluates to false, another set of actions is performed (this can be optional). We can represent a branch using both pseudocode and a flowchart, as shown in Figures 1.25 and 1.26.

> **Link**
>
> To learn more about types of branches, including IF, THEN, ELSE and ELSEIF (ELIF), see Branches, in Control structures, in *Unit 4: Software Design and Development Project*.

```
IF age >= 18 AND validDebitCard = true THEN
  OUTPUT "Holiday booked!"
ELSE
  OUTPUT "Cannot book holiday at this time!"
ENDIF
```

Figure 1.25: Representing a branch using pseudocode

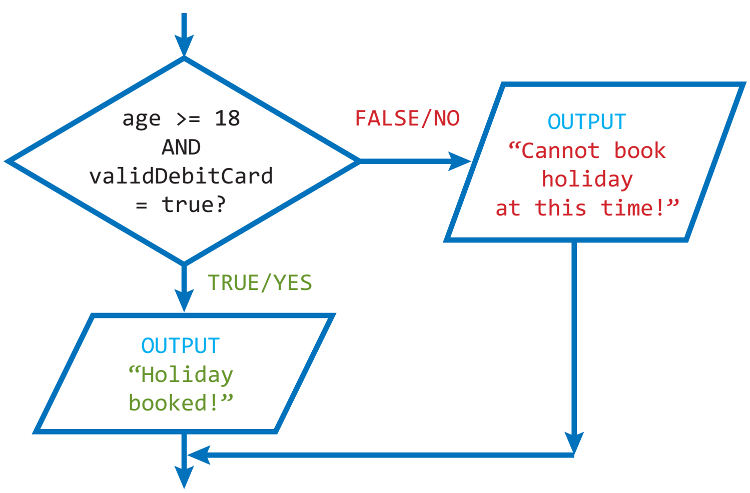


Figure 1.26: Representing a branch using a flowchart

Sometimes, we may wish to have multiple branches within the same part of the algorithm, particularly when there are several TRUE possibilities. For example, as part of a company's online loyalty scheme, customers are awarded bronze, silver or gold discounts depending on the value of goods they have previously purchased.

<£50 No discount

£50 - £100 bronze

£101 - £200 silver

£201+ gold

You can use a series of IF...THEN...ELSEIFs to handle this situation, as shown in Figures 1.27 and 1.28.

```
IF valueGoods < 50 THEN
   OUTPUT "No discount!"
ELSEIF valueGoods < 101 THEN
   OUTPUT "Bronze discount"
ELSEIF valueGoods < 201 THEN
   OUTPUT "Silver discount"
ELSE
   OUTPUT "Gold discount"
ENDIF
```

Figure 1.27: IF...THEN...ELSEIFs using pseudocode to represent the branch

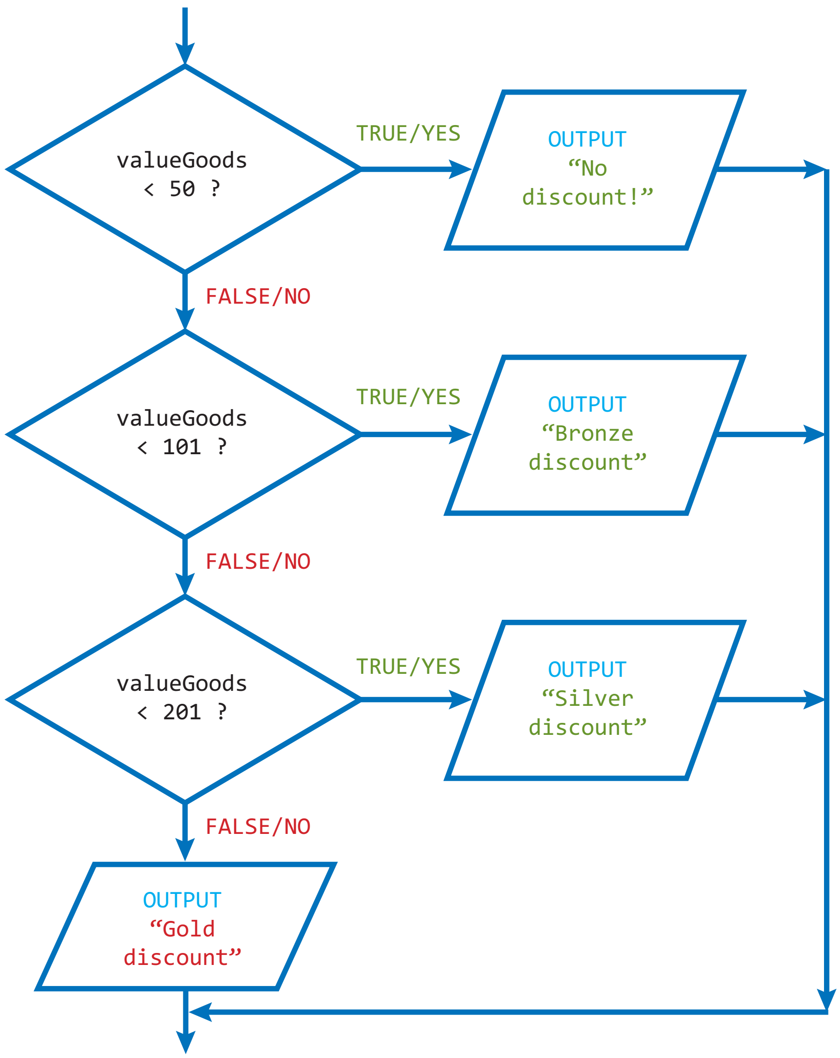


Figure 1.28: IF...THEN...ELSEIFs using a flowchart to represent the branch

In programming languages such as Python®, ELIF can be used to offer multiple alternate branches, as shown in Figure 1.29.

```
# Branch example program in Python

#!/usr/bin/python

valueGoods = 117;

if valueGoods < 50:
   print("No discount")
elif valueGoods < 101:
   print("Bronze discount")
elif valueGoods < 201:
   print("Silver discount")
else:
   print("Gold discount")
```

Figure 1.29: Python® ELIF branch example

Function calls

A function is a separate block of code that performs a specific job. Some programming languages define a function by stating that it should return a value of some kind, differentiating it from a procedure which merely performs an action. Many languages, particularly those in the C family, make no such distinction – they are simply 'functions'.

Functions, also called modules, subroutines or procedures, depending on the programming language, are responsible for performing a single task and are typically somewhere between 5 and 50 lines of code long.

The use of such functions means that code tends to be easier to write, easier to read and debug, reusable through multiple solutions and allows a single application to be divided and worked on by several programmers simultaneously (saving development time).

Figure 1.30 shows modules (functions) being called from a main program.

Key term

Function – a separate block of code which performs a specific job.

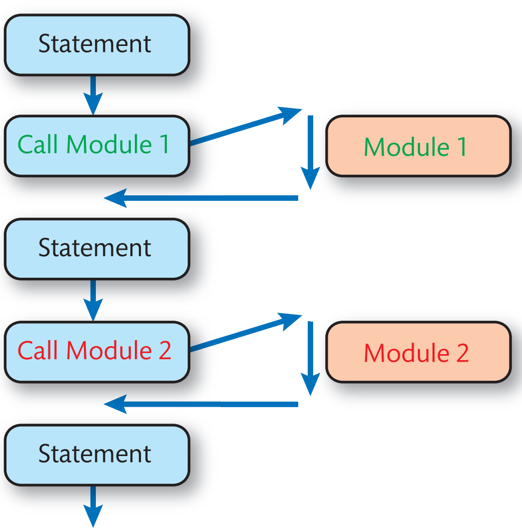


Figure 1.30: Modules (functions) being called from a main program

Program flow follows line by line until a function is called. When this happens, control is passed to the module which is then also executed line by line. At the end of the module, control is passed back to the line of code after the original function call. This pattern repeats until the end of the program. It is very common for the same functions

to be called multiple times in the life of a working program. Functions can even call other modules as necessary. Functions which are part of the programming languages are called library or built-in functions. Functions created by a programmer are called user or programmer-defined.

The C# code shown in Figure 1.31 demonstrates the use of a programmer-defined function to calculate the area of a circle, given a specific radius.

```
class Program
{
    const float PI = 3.14f;

    static void Main(string[] args)
    {
        float radius;
        float area;

        radius = 10.0f;
        area = calcAreaCircle(radius);  // function call

        Console.WriteLine("Area of circle is {0}", area);
        Console.ReadKey();
    }

    //function declaration
    public static float calcAreaCircle(float radius)
    {
        return PI * radius * radius;
    }
}
```

Figure 1.31: C# programmer-defined function

Declaration

Functions are usually defined (declared) in a separate section of the program's code, possibly even in a separate file (depending on the programming language). In the C# example shown in Figure 1.31, the function is declared at the end of the main function's code.

Arguments and formal paramaters

It is commonly accepted that the function's **arguments** are the values (variables, constants or literals) that are passed *into* the function to be processed. In Figure 1.31 this would be the 'radius' variable which is declared and initialised as a float variable with a value of 10 in the main function.

The function declaration will include not only the function's name ('calcAreaCircle') but also the **formal parameters** expected ('radius') and the return type of the function (it returns a float). Arguments and formal parameters do not need to have the same name but they typically have the same quantity, data type and order: that is, if a function expects three arguments (for example, two integers and a float) then these values must be passed to it for it to work correctly.

Calling functions

A function is called by using its name and, if needed, arguments are passed to it to be processed. The result is stored in a variable which is of the same data type as the function's return type.

Data structures

A data structure is a technique used to collect and organise data items into a formal structure, which allows more efficient processing. Although the availability of certain data structures varies between different programming languages, many are very common. These include strings, arrays (one and two dimensional), stacks, queues and records.

Sometimes, it is possible to program more efficiently through the selection and use of specific data structures, especially when combined with iteration (loop) control structures. Software developers need to become familiar with many data structures as they learn about different programming languages.

A data structure is used to store a collection of characters with one character minimally requiring one byte of RAM.

String (or text)

Strings may be of fixed length (e.g. 10 characters long) or use a special 'terminator' character to mark their end. This may mean that a string will require an additional character but it allows them to be dynamic in length.

Planet

0	1	2	3	4	5
E	a	r	t	h	#

Each individual character is accessible using its positional index, for example **Planet[2] is 'r'**, as seen in the sample C# code extract in Figure 1.32.

```
string Planet;
Planet = "Earth";

Console.WriteLine("Whole string is {0}", Planet);
Console.WriteLine("3rd character is {0}", Planet[2]);
```

Figure 1.32: Working with C# strings

This can be used to store simple text entered by the user, for example a username.

Strings may also be used to enter more complex text for processing, for example the content of an email, an instant message or a tweet.

Some languages may use a static one-dimensional array of characters to simulate a string.

Array (one-dimensional)

Classically, this is a static data structure (having a fixed size), but modern programming languages are generally more flexible, allowing an array to be dynamically resized as needed.

Typically, an array can store only one type of data, but any type of data (integers, characters, decimal, Booleans etc.) is acceptable.

For example, if you wanted to store 7 daily maximum temperatures (in degrees Celsius) for a local weather station, you could create an array of 7 decimal numbers.

Temperatures

0	1	2	3	4	5	6
12.50	10.45	12.30	14.60	17.70	11.20	12.50

Tip

Almost all programming languages have library functions which are freely available for the programmer to use when solving complex problems. These allow the programmer to perform common (but crucial) tasks on data, for example formatting its appearance, finding the length of a string or the square of a number. It is also possible to download and install third-party functions from reputable websites to expand the programming language.

Link

To learn more about how to define, declare and call functions, see Function calls, in Control structures, in *Unit 4: Software Design and Development Project*.

This may appear similar to a string (indeed, you can think of a string as an array of characters) and, again, it is possible to access individual elements (or items) in the array by using the required index, for example Temperatures[4] is 17.70.

Figure 1.33 demonstrates the creation and access of a simple one-dimensional array in C#.

```
//create the array of decimal temperatures
float[] temperature = new float[7];

//initialise each element
temperature[0] = 12.50f;
temperature[1] = 10.45f;
temperature[2] = 12.30f;
temperature[3] = 14.60f;
temperature[4] = 17.70f;
temperature[5] = 11.20f;
temperature[6] = 12.50f;

//select Thursday's and output it
Console.WriteLine("Temperature on Thursday is {0} degrees C", temperature[4]);

//wait for keypress to continue
Console.ReadLine();
```

Figure 1.33: C# one-dimensional zero-based array

Array (two-dimensional)

These data structures are similar to one-dimensional arrays but store multiple rows of data.

For example, if you wanted to store 7 daily maximum temperatures (in degrees Celsius) for a local weather station over three consecutive days, you could create a two-dimensional array of 7 by 3 decimal numbers.

Temperatures

	0	1	2	3	4	5	6
0	12.50	10.45	12.30	14.60	17.70	11.20	12.50
1	12.22	11.00	10.00	20.00	21.00	14.00	15.50
2	13.00	14.00	14.50	14.60	12.30	14.00	14.60

Once again, it is possible to access individual elements (or items) in the array by using both indices, for example Temperature[4][1] is 21.00.

Higher orders of dimensions are possible. For example, a three-dimensional array could be used to store 21 temperatures for 5 different weather stations.

Tip

The actual order of indices may vary between different programming languages. The general rule is that the element access order will reflect the declaration order of the array.

Record (or structure)

A record is a similar concept to an array but differs because it can store a mix of data types within its structure.

For example, storing a student's details.

Student

StudentID	Forename	Surname	Age	Enrolled
2001	Preston	Myla	21	True

In this example, StudentID and Age are integers, Forename and Surname are strings and Enrolled is a Boolean. A C# implementation of a simple record structure is shown in Figure 1.34.

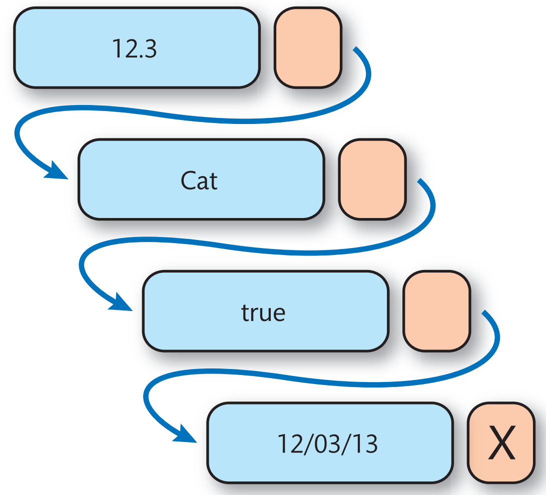
```
using System;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    public class testStructure
    {
        //create the record structure
        struct Student
        {
            public int StudentID;
            public string Surname;
            public string Forename;
            public int Age;
            public bool Enrolled;
        };

        public static void Main(string[] args)
        {
            Student mystudent;    //declare mystudent of type Student structure

            //initialise the elements in the record structure
            mystudent.StudentID = 2001;
            mystudent.Surname = "Preston";
            mystudent.Forename = "Myla";
            mystudent.Age = 21;
            mystudent.Enrolled = true;

            //print the data in the record structure
            Console.WriteLine("ID        : {0}", mystudent.StudentID);
            Console.WriteLine("Surname   : {0}", mystudent.Surname);
            Console.WriteLine("Firstname : {0}", mystudent.Forename);
            Console.WriteLine("Age       : {0}", mystudent.Age);
            Console.WriteLine("Enrolled  : {0}", mystudent.Enrolled);

            //wait for a keypress to continue
            Console.ReadKey();
        }
    }
}
```

Figure 1.34: C# simple record structure using struct

It is also possible to create an array of structures to store multiple instances of a record. For this reason, a record (or struct) can be seen as the basis for simple record storage in a database table or a binary data file.

Lists

A list (usually called a **linked list**) has a number of data items connected together in a 'chain' using **pointers**.



▶ **Figure 1.35:** Linked list of four data items

Figure 1.36 shows the code for the linked list.

```
using System.IO;
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        // Create a new LinkedList.
        LinkedList<String> ll = new LinkedList<String>();

        // Create a new LinkedListNode of type String
        LinkedListNode<String> lln = new LinkedListNode<String>( "Rebekah" );

        // Add the "Rebekah" node to the empty linked list
        ll.AddLast( lln );

        // Add more nodes to the linked list, but specify where...
        ll.AddFirst( "Josie" );
        ll.AddLast( "Christopher" );
        ll.AddFirst( "Matthew" );

        //display the linked list using a for loop

        Console.WriteLine("We currently have {0} student(s) stored.", ll.Count);
        foreach (var student in ll)
        {
            Console.WriteLine("Node's data is {0}", student);
        }
    }
}
```

▶ **Figure 1.36:** C# simple linked list using built-in linked list class

Each **pointer** connects the next data item in the list.

Unlike a typical array, each data item may be a different data type. The last pointer has a NULL value to show the end of the list.

Key term

Pointer – this is a special type of variable which stores the memory address of another variable in the RAM. This allows the second variable to be accessed indirectly by referencing the first variable. Pointers are used in more advanced programming to make solutions more efficient but can be dangerous if not used correctly.

The logical order of the data items does not have to match its physical order in the RAM. Linked lists are fast, flexible alternatives to using an array.

After running this code, the output shown in figure 1.37 should be displayed.

Figure 1.37: Sample output of linked list

As you can see, C# has inserted each node into the requested location in the list. It is also possible to insert a new node between existing nodes.

Sets

Sets can be used to organise data and define the interrelationships between different sets of data. This allows you to easily search and **filter** potentially complex data. Some programming languages have data types which support set-style functionality and this logic can be used to form user permissions.

Key terms

Set – a collection of distinct objects. Sets can contain anything (e.g. names, numbers, colours or letters of the alphabet) and may consist of many different members.

Filter – by using a filter you can include or exclude certain values when running a search.

For example, you may have two groups of people with different access rights in a particular application.

Set A is the Admin group, which contains John, Ahmed and Jo.

This can be expressed as Set A = {John, Ahmed, Jo}.

Set B is the Finance group, which contains Claire, Jo, Niamh and Phil.

This can be expressed as Set B = {Claire, Jo, Niamh, Phil}.

You can represent these sets of data as a Venn diagram, as shown in figure 1.38.

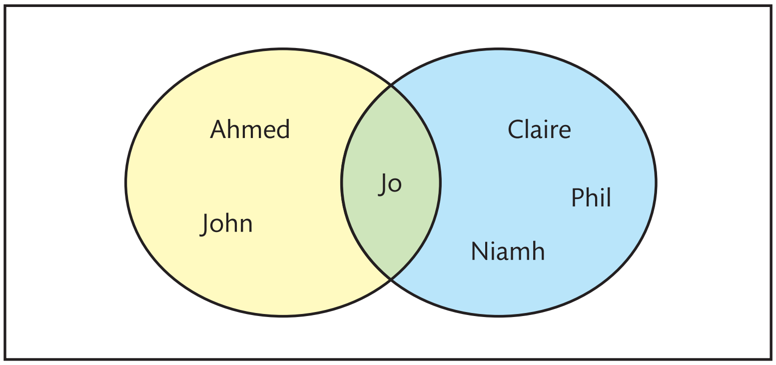


Figure 1.38: Venn diagram

Figure 1.39 demonstrates how programming languages such as C# can declare this type of set data and interrogate it.

```
using System.IO;
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        //declare and initialise each set
        HashSet<string> admin = new HashSet<string>() {"John","Ahmed","Jo"};
        HashSet<string> finance = new HashSet<string>() {"Jo","Claire","Niamh","Phil"};

        //display the admin set
        Console.WriteLine("admin set contains {0} member(s): ", admin.Count);
        foreach (string member in admin)
        {
            Console.WriteLine("{0}", member);
        }

        //display the finance set
        Console.WriteLine("finance set contains {0} member(s): ", finance.Count);
        foreach (string member in finance)
        {
            Console.WriteLine("{0}", member);
        }

        //Create a new empty HashSet for everyone
        HashSet<string> everyone = new HashSet<string>(admin);

        //perform a union operation
        everyone.UnionWith(finance);

        //display the union
        Console.WriteLine("Both sets contain {0} member(s): ", everyone.Count);
        foreach (string member in everyone)
        {
            Console.WriteLine("{0}", member);
        }

        // Create a new empty HashSet for just members belong to both groups
        HashSet<string> both = new HashSet<string>(admin);

        //perform an intersect operation
        both.IntersectWith(finance);

        //display the intersect
        Console.WriteLine("Members that occur in both sets: {0}", both.Count);
        foreach (string member in both)
        {
            Console.WriteLine("{0}", member);
        }
    }
}
```

Figure 1.39: C# using sets to solve a problem

There is a particular function in this example application that can only be accessed by administrators (members of the Admin group) who work in the Finance department (members of the Finance Group). You can use the intersection of the sets to find out who this person is.

```
admin set contains 3 member(s):
John
Ahmed
 Jo
finance set contains 4 member(s):
Jo
Claire
Niamh
Phil
Both sets contain 6 member(s):
John
Ahmed
 Jo
Claire
Niamh
Phil
Members that occur in both sets: 1
 Jo
```

Figure 1.40: Sample output of set example

Looking at Figure 1.40, you can easily see that the only user who exists in both sets is Jo. This means that only Jo has access to this function.

All algorithms rely on combinations of the control structures that you have already encountered – loops, branches, function calls, etc.

Link

To learn more about some of the most common data structures (lists, arrays, records and sets), see Data structures, in *Unit 4: Software Design and Development Project*.

Common/standard algorithms

Developing programs successfully often makes use of standard types of algorithm to store and process data. Although the real-life context of a problem may be different (for example, insurance, education, retail) the techniques that are required are often generic. For example, the need to sort a list of names into alphabetical order could occur in many different business situations.

For this reason, it is a good idea to become familiar with a range of different types of algorithm that will form part of your problem-solving toolkit; that is, techniques you can rely on and apply to new situations.

Common or standard algorithm types include those for sorting, searching, counting and validation, as well as stacks and queues to implement sorting and searching.

Using stacks and queues to implement sorting and searching

Stack and queue algorithms are used to implement sorting and searching algorithms.

Stack (LIFO)

A stack is known as a **LIFO** data structure and has two basic operations: push and pull (sometimes known as pop).

Stacks are a vital part of any computer platform's operating system and are a common tool for the programmer when developing solutions.

Key term

LIFO – last in, first out – a way of describing how data is treated in some data structures; the last item of data pushed on is also the first item of data that may be pulled back off.

Stacks are conceptually viewed vertically as shown in Figure 1.41.

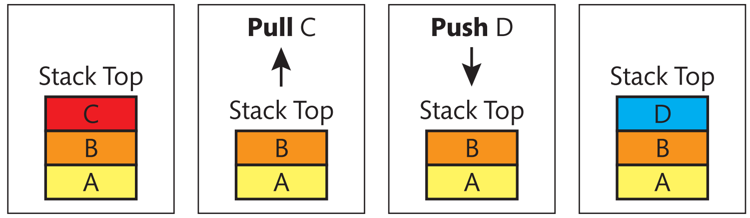


Figure 1.41: Stack operations

Figure 1.42 shows a stack operation in C#.

```
using System;
using System.Collections;

namespace stack
{
    class Program
    {
        static void Main(string[] args)
        {
            char data;                  //single item of stack data
            Stack st = new Stack();     //our stack

            //push onto stack
            st.Push('A');
            st.Push('B');
            st.Push('C');

            //show the stack
            Console.WriteLine("Current stack: ");
            foreach (char ch in st)
            {
                Console.WriteLine(ch);
            }

            //remove from stack top - it should be "C"
            data = (char)st.Pop();
            Console.WriteLine("The popped value: {0}", data);

            //push onto stack again
            st.Push('D');

            //show the changed stack
            Console.WriteLine("Current stack: ");
            foreach (char ch in st)
            {
                Console.WriteLine(ch);
            }

            //wait for keypress to continue
            Console.ReadKey();
        }
    }
}
```

Figure 1.42: Stack operation in C#

Unlike an array where elements may be accessed in any order, a stack can only be accessed in a last in, first out manner. This can be particularly useful, especially when processing recursive algorithms. Stacks are often included in a programming language's library, complete with the necessary methods to process them.

Key term

FIFO – first in, first out – a way of describing how data is treated in some data structures; the first item of data added is also the first item of data that may be removed.

Queues

In contrast, a simple queue is known as a **FIFO** data structure and has two basic operations.

FIFO

This adds (to the tail) or 'enqueues' the data and removes (from the head) or 'dequeues' data. Only data at the head of the queue can be accessed and removed.

Queues, like stacks, are a vital part of any computer platform's operating system. For example, you are likely to be familiar with the concept of a printer queue. For the programmer, they are an excellent way of managing task processing.

Queues are conceptually viewed horizontally as shown in Figure 1.43.

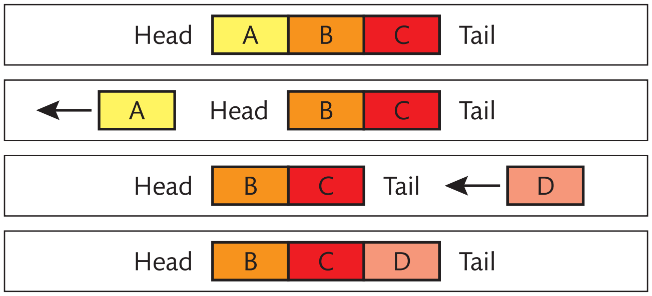


Figure 1.43: Queue operations

Figure 1.44 demonstrates the creation and access of a simple queue in C#.

```
using System;
using System.Collections;
using System.Linq;
using System.Text;

namespace queue
{
    class Program
    {
        static void Main(string[] args)
        {
            char data;              //single item of queue data
            Queue q = new Queue();  //our queue

            //add to queue
            q.Enqueue('A');
            q.Enqueue('B');
            q.Enqueue('C');

            //show the queue
            Console.WriteLine("Current queue: ");
            foreach (char ch in q)
            {
                Console.Write(ch + " ");
            }

            //remove from head - it should be 'A'
            data = (char)q.Dequeue();
            Console.WriteLine("The removed value: {0}", data);

            //add to queue again
            q.Enqueue('D');

            //show the changed queue
            Console.WriteLine("Current queue: ");
            foreach (char ch in q)
            {
                Console.Write(ch + " ");
            }

            //wait for keypress to continue
            Console.ReadKey();
        }
    }
}
```

Figure 1.44: C# simple queue

Sorting

Sorting is a basic process whereby items are ordered based on a chosen attribute, for example ordering videogames alphabetically by their title or albums in a media library by the band or artist's name.

In computer science, many different sorting algorithms exist, each offering various levels of complexity and efficiency that make some more suited to certain situations, for example sorting small or large sets of data quickly.

The three most common sorting algorithms are:

- bubble sort
- insertion sort
- quicksort.

You will now look at each sorting algorithm in turn to identify how they work and their strengths and weaknesses.

Bubble sort

The bubble sort is probably the easiest sorting algorithm to understand. As you may have guessed, its name mirrors the way that bubbles float to the surface of a liquid. In a bubble sort, a data item is physically moved through a list until it reaches its correctly sorted position. This is achieved by comparing each neighbouring pair and swapping their position as necessary. This process can require many complete '**passes**' along the data until all data items are in their correctly sorted positions.

A simple example of using a bubble sort algorithm to sort an array of five random integers into ascending order (small to large) is shown in Table 1.7.

Table 1.7: Bubble sort example

First Pass	Second Pass	Third pass	Fourth pass (0 swaps)
(9, 2, 3, 1, 6) -> (2, 9, 3, 1, 6)	(2, 3, 1, 9, 6) -> (2, 3, 1, 6, 9)	(2, 1, 3, 6, 9) -> (1, 2, 3, 6, 9)	(1, 2, 3, 6, 9) -> (1, 2, 3, 6, 9)
(2, 9, 3, 1, 6) -> (2, 3, 9, 1, 6)	(2, 3, 1, 9, 6) -> (2, 1, 3, 6, 9)	(1, 2, 3, 6, 9) -> (1, 2, 3, 6, 9)	(1, 2, 3, 6, 9) -> (1, 2, 3, 6, 9)
(2, 3, 9, 1, 6) -> (2, 3, 1, 9, 6)	(2, 1, 3, 6, 9) -> (2, 1, 3, 6, 9)	(1, 2, 3, 6, 9) -> (1, 2, 3, 6, 9)	(1, 2, 3, 6, 9) -> (1, 2, 3, 6, 9)
(2, 3, 1, 9, 6) -> (2, 3, 1, 6, 9)	(2, 1, 3, 6, 9) -> (2, 1, 3, 6, 9)	(1, 2, 3, 6, 9) -> (1, 2, 3, 6, 9)	(1, 2, 3, 6, 9) -> (1, 2, 3, 6, 9)

You should be able to see the larger values moving (bubbling) to the right and the smaller values to the left after each pass. In the simplest bubble sort algorithm, it is common for the data to become fully sorted *before* the algorithm is able to detect this. The algorithm will only stop when a full pass is completed with no swap occurring (see the fourth pass). This is usually achieved through the use of a Boolean 'swapped' flag, as shown in Figure 1.45, which shows the bubble sort algorithm written as pseudocode.

```
BEGIN
  size = length (data)
  REPEAT
   swapped = false
   FOR index = 1 TO size -1
     IF data[index + 1] < data [index] THEN
       swap data[index + 1], data[index]
       swapped = true
     ELSE
       no swap needed
     ENDIF
    ENDFOR
  UNTIL swapped = false
END
```

Figure 1.45: A bubble sort algorithm

Although various tweaks to the algorithm can improve its efficiency, bubble sorts are not the fastest sorting method and should be avoided when dealing with large data sets.

Insertion sort

This type of sorting algorithm is similar to a bubble sort as it sorts 'in-place'. However, its algorithm is very different from that of the bubble sort and involves maintaining a sub-list within the unsorted data, which is always kept sorted.

Traditionally, this is achieved in the lower part of an array with newly selected elements being inserted into their correct position. A simple example of using an insertion sort algorithm to sort an array of five random integers into ascending order (small to large) is shown in Table 1.8.

Table1.8: Tracing through an insertion sort algorithm

Action	Data (sub-list)	Decision
Unsorted array of data	(1,10,4,5,19)	
Compare element 0 and 1	(1,10,4,5,19)	No change, put 1into sorted sub-list
Compare element 1 and 2	(1,10,4,5,19)	10 is not in correct position
Swap element 1 and 2	(1,4,10,5,19)	
Check element 0 and 1 of sub-list	(1,4,10,5,19)	1 and 4 are already in the correct order, so leave sub-list alone
Compare element 2 and 3	(1,4,10,5,19)	10 is not in correct position
Swap element 2 and 3	(1,4,5,10,19)	
Check elements 0, 1 and 2 of sub-list	(1,4,5,10,19)	1, 4 and 5 are already in the correct order, so leave sub-list alone
Compare element 3 and 4	(1,4,5,10,19)	10 is in the correct position
Check elements 0,1,2 and 3 of sub-list	(1,4,5,10,19)	1, 4, 5 and 10 are already in the correct order, so leave sub-list alone
No comparison; it is the last element	(1,4,5,10,19)	Array should be sorted

Research

Modify the original data set to (1, 10, 5, 4, 19) and trace the insertion sort algorithm to see what would happen.

Figure 1.46 shows the insertion sort algorithm written as pseudocode.

```
BEGIN
  size = length (data)
  hole = 0
  insertValue = 0

  FOR index = 0 TO size -1

    insertValue = data[index]
    hole = index

    WHILE hole > 0 AND data[index - 1] > insertValue
      swap data[hole], data[hole-1]
      decrement hole
    ENDWHILE

    Data[hole] = insertValue

   ENDFOR
END
```

Figure 1.46: An insertion sort algorithm

Similarly to the bubble sort, the insertion sort algorithm is not suitable for larger data sets.

Key term

Recursion – this is an important concept in computer science and occurs when the success of solving a larger problem relies on solving smaller copies of that problem first. In programming, recursion is typically achieved when a function calls itself a number of times until a terminating condition is met and each result is returned to its parent function until the final result is attained.

Quicksort

Of the three sorting algorithms we are focusing on, the quicksort, invented by C.A.R. Hoare, is probably the most complex to understand. The reason for its complexity is that it involves a process known as **recursion**.

As the name suggests, the quicksort's main selling point is that, compared with other sorting algorithms, it is considerably faster in most instances, especially when dealing with larger data sets. This is because it uses a 'divide and conquer' approach by using two phases:

1. the **partition** phase – which works out how to divide the data into two partitions
2. the **sort** phase – which does the actual sorting of each smaller partition.

Below is the pseudocode for each phase. Figure 1.47 shows the pseudocode for the quicksort phase. Take note of the recursive call to sort each half.

```
FUNCTION quicksort (data, start, end)
BEGIN
  IF start < end
    pivot = partition(data, start, end)
    CALL quicksort (data, start, pivot - 1)
    CALL quicksort (data, pivot + 1, end)
  ENDIF
  RETURN data
END
```

Figure 1.47: Recursive quicksort phase

Figure 1.48 shows the pseudocode for the partition phase.

```
FUNCTION partition (data, start, end)
BEGIN
  pivot = data[start]
  left = start+1
  right = end
  done = false
  WHILE done == false
    WHILE left <= right AND data[left] <= pivot
      left = left + 1
    ENDWHILE
    WHILE data[right] >= pivot and right >=left
      right = right -1
    ENDWHILE
    IF right < left THEN
      done = true
    ELSE
      temp = data[left]
      data[left] = data[right]
      data[right] = temp
    ENDIF
  ENDWHILE
  temp = data[start]
  data [start] = data[right]
  data [right] = temp
  RETURN right
END
```

Figure 1.48: Partition phase

Most modern programming languages have pre-written libraries that contain standard functions for sorting data stored in arrays and lists. Despite this, having a working knowledge of the processing involved often proves invaluable when solving more complex problems.

Searching

Another basic algorithm enables data to be searched for a particular value. There are two basic types of searching algorithm: serial/linear and binary search.

Serial/linear search

A linear search of a specific value within a list of data items is a reasonably simple algorithm to understand and implement. It is perhaps one of the most frequently used algorithms in a developer's toolkit.

This algorithm needs to 'walk' along the data (usually stored in an array), compare the element against the search value and, if they are identical, return the index position where the match has been made. If all data items have been checked and no match has occurred, an error should be flagged. This process is shown in Table 1.10.

Table 1.10: Linear search with successful and unsuccessful outcomes

Search value	Data items	Linear search result
3	0 1 2 3 4 (9, 2, 3, 1, 6)	2
5	0 1 2 3 4 (9, 2, 3, 1, 6)	–99

Care should be taken not to return a 0 if no match is found; if the array is zero-based then 0 would in fact be the first element. For this reason, using a **sentinel value** (something unusual, for example –99) may be considered a good replacement. At first glance, you may have thought that using a Boolean false seems a good choice, but care should be taken, as false usually equates to 0 (which could again be a valid position depending on the array). Figure 1.49 shows the linear search algorithm written as pseudocode.

```
BEGIN
  INPUT search
  size = length (data)
  found = -99
  FOR index = 0 TO size-1
    IF data[index] = search THEN
      found = index
      BREAK
    ENDIF
  ENDFOR
  OUTPUT found
END
```

Figure 1.49: Linear search algorithm using a sentinel

Linear searches are improved by pre-sorting the data but the impact of performing a double operation (sorting then searching) should be considered. Processing problems often occur when there is a large quantity of data, especially if the search value is found towards the end of the list (or not at all, having checked every single one). An alternative to a linear search is a binary search.

Binary search

Unlike a linear search, the binary search has to work on a sorted list of data.

Once the data is successfully sorted, the algorithm picks the middle value in the list. Then it compares this value to the search value. If the search value is smaller, it discards the values above the middle value; if the search value is bigger, it discards the values below. It repeats this process with increasingly smaller blocks of data until the search value is found or established to not be in the list. This process is shown in Figure 1.50.

Search value	Unsorted data
10	0 1 2 3 4 5 6 7 8 9 (25,13,8,7,6,5,20,19,**10**,17)
	Sorted data
	0 1 2 3 4 5 6 7 8 9 (5,6,7,8,**10**,13,17,19,20,25)
	Find midpoint in data
	0 1 2 3 4 5 6 7 8 9 (5,6,7,8,**10**,13,17,19,20,25) Is 10 < 13? **Yes**, so discard 5–9 0 1 2 3 4 (5,6,7,8,**10**)
	Find midpoint in data
	0 1 2 3 4 (5,6,7,8,**10**) Is **10** < 7? **No**, so discard 0–2
	Only two are left; is our search value one of them?
	3 4 (8,**10**) **Yes**, **10** is in position **4**.

Figure 1.50: Binary search

As you can see, the binary search method has only required three comparisons in total. If you examine the original unsorted data, a linear search would have needed to perform nine comparisons to find it in location 8. Even if you had performed the linear search on the sorted data, it still would have required five comparisons to find it in position 4. It would appear, therefore, that the binary search is more efficient, although you would have to include the time taken to sort the data first before making a true comparison.

Figure 1.51 shows a simplified algorithm for this type of search.

```
BEGIN

  data = array of integers
  sortedData = Sort(data)
  size = size of sortedData
  found = false
  midPoint = 0

  INPUT search

  lowerBound = 0
  upperBound = size-1

  WHILE found = false

   IF upperBound < lowerBound THEN
     BREAK
   ENDIF

   midPoint = lowerBound + (upperBound - lowerBound) / 2

   IF sortedData[midPoint] < search THEN
     lowerBound = midPoint + 1
   ENDIF

   IF sortedData[midPoint] > search THEN
     upperBound = midPoint – 1
   ENDIF

   IF sortedData[midPoint] = search THEN
     found = true
     BREAK
   ENDIF
  ENDWHILE

  IF found = false THEN
   OUTPUT search " was not found."
  ELSE
   OUTPUT search " was found in location " midPoint
  ENDIF

END
```

Figure 1.51: Simplified binary search algorithm

Other standard algorithms

There are two other types of standard algorithm that will be looked at in this section. These are count occurrences and input validation.

Count occurrences

This is a variation of the linear search algorithm. However, rather than confirming the match, you are interested in keeping a count of how many times a specific value is found in a set of data. This counter is then output after the loop finishes.

Much of the pseudocode can be adapted from the linear search, as shown in Figure 1.52.

```
BEGIN
  INPUT search
  size = length (data)
  count = 0
  FOR index = 0 TO size-1
    IF data[index] = search THEN
      increment count
    ENDIF
  ENDFOR
  OUTPUT search " was found " count " time(s)."
END
```

Figure 1.52: Typical count occurrences algorithm

Input validation

Validation techniques vary greatly depending on the type of validation required, for example whether these are for checking the type of data being entered or whether the data entered is in a desired range. Most validation techniques use a combination of looping and comparison.

Figure 1.53 shows a typical range check algorithm written as pseudocode.

```
BEGIN
  min = 1
  max = 10
  REPEAT
    OUTPUT "Enter a value between " min " and " max
    INPUT value
    IF value < min OR value > max THEN
       OUTPUT "Value must be between " min " and " max
    ENDIF
  UNTIL value >= min AND value <= max
END
```

Figure 1.53: Typical range check algorithm

PAUSE POINT Can you explain what the assessment outcome was about? What elements did you find easiest?

Hint Close the book and think about the different operators, control structures, data structures, algorithms and functions that are used to build a program.

Extend You have seen a number of different search and sort algorithms; why do you think there are different methods for achieving the same result – how do they differ?

D Types of programming and mark-up languages

Different types of problem have spawned different programming styles. Each style, known as a paradigm, aims to solve a problem in a different way, often to fulfil different user needs.

There are three main types of computer programming – procedural, object oriented and event driven – and each solves problems in a different way and therefore has different uses. When coding for the web, different issues come into play in terms of user needs and this section will also look at the type of programming and mark-up languages used to code websites. In this section, you will explore the issues surrounding the translation of code between programming languages.

Procedural programming

Procedural languages are often the first ones learned by a programmer. They are often considered to be a general-purpose tool and are used to create many different types of application.

Common procedural programming languages include:

- C
- Perl
- Python®.

Procedural programs are typically written as a series of well-defined steps which solve a set problem, for example performing a simple calculation in C, as shown in Figure 1.54.

```
#include <stdio.h>
#include <string.h>

int main()
{
  int a;
  int b;
  int c;

  puts("Enter first number");
  scanf("%d", &a);
  puts("Enter second number");
  scanf("%d", &b);

  c = a + b;
  printf("%d + %d = %d",a,b,c);

  return 0;
}
```

Figure 1.54: C program code displaying the sum of two inputted numbers

You need to be able to recognise the structure of a procedural program, identify its major components by name and state their purpose.

Figure 1.55 shows a typical program written in C, demonstrating the common procedural features that you should be able to identify, interpret and use correctly. As you will notice, various aspects of the program have been tagged with descriptions to help you to analyse the code.

Link

For more about types of programming, see *Unit 4: Software Design and Development Project*, *Unit 15: Website Development* and *Unit 17: Mobile Apps Development*.

```
#include <stdio.h>

#define MAXSIZE 5

//create a Boolean data type
typedef enum {false,true} bool;


//global data
int data[MAXSIZE];
int stackTop;

//function prototypes
bool stackFull(void);
bool stackEmpty(void);
char showMenu(void);
void pushData(void);
void popData(void);
void showStack(void);
void clearBuffer(void);

int main()
{
  //local data
  char menu;
  stackTop = -1;

  //main loop
  do
  {
    menu = showMenu();
    switch (menu)
    {
      //push chosen
      case '1' : if (!stackFull())
            {
              pushData();
            }
            else
            {
              puts("Sorry, stack is full.\n");
            }
            clearBuffer();
            break;

      //pop chosen
      case '2' : if (!stackEmpty())
            {
              popData();
            }
            else
            {
              puts("Sorry, stack is empty.\n");
            }
            clearBuffer();
            break;
```

Built-in libraries

Constant

Global variables

Functions and procedures

Local variables

Function call

Function calls

Conditional branches

Function calls

Conditional branch

```
      //show stack chosen
      case '3' : if (!stackEmpty())
            {
              showStack();
            }
            else
            {
              puts("Sorry, stack is empty.\n");
            }
            clearBuffer();
            break;
    }

  } while (menu != 'X' && menu !='x');

  puts("Goodbye!");

  return 0;
}

//show the stack simulator menu and get option
char showMenu (void)
{
  char option;

  puts("\n\nStack Simulator");
  puts("1 - Push data");
  puts("2 - Pop data");
  puts("3 - Show stack");
  puts("X - eXit");
  puts("Enter choice: 1,2,3 or X:");
  scanf("%c",&option);
  return option;
}

//is the stack empty?
bool stackEmpty(void)
{
  if (stackTop==-1)
  {
    return true;
  }
  else
  {
    return false;
  }
}
//is the stack full?
bool stackFull(void)
{
  if (stackTop==(MAXSIZE-1))
  {
    return true;
  }
  else
  {
    return false;
  }
}
```

Function calls

Conditional branch

Iteration

Local variable

Sequence of output statements

Input statement

Function

Conditional branches

Function

Conditional branches

```c
//push data onto the stack
void pushData(void)
{
  int newData;
  puts("Enter new Data");
  scanf("%d",&newData);
  stackTop++;
  data[stackTop] = newData;
}

//pop data from the stack
void popData(void)
{
  int poppedData;
  poppedData = data[stackTop];
  printf("Popped data is %d",poppedData);
  stackTop--;

}
//display the stack
void showStack(void)
{
  puts("Stack Top");
  for (int counter=stackTop; counter>=0;counter--)
  {
    printf("%d is %d\n",counter,data[counter]);
  }
}

//clear keyboard buffer of any stray return/line feeds
void clearBuffer(void)
{
  while (getchar() != '\n');
}
```

Procedure

Sequence of statements

Procedure

Sequence of statements

Procedure

Iteration

Procedure

Iteration

Figure 1.55: C program using many common procedural features

As you may have noticed, the C program shown in Figure 1.57 is a full stack simulator, which is coded in a procedural fashion, suitable for execution on various computer platforms (for example Microsoft® Windows® operating system and Linux).

At this level of study, you should be able to interpret programs of this complexity (or greater), be comfortable with the challenge of identifying the different components of a procedural program and recognise how they connect to form a fully working solution.

Object-oriented programming

Object-oriented (OO) programming is a popular modern approach to programming. OO languages rely on the concepts of 'classes' and 'objects' to solve real-world problems. Due to this approach, they are a popular choice for designing videogames, e-commerce websites, database systems and user interfaces.

Common object oriented programming languages include:

- C++
- Microsoft® C#
- Oracle® JavaScript
- PHP
- Ada.

In OO, objects are created from classes, which are usually modelled on real-world 'things' such as customers, bank accounts, products, orders etc. Each class acts as a software blueprint that encapsulates (or contains) the thing's state (its data or properties) and behaviour (its functions or methods) in program code (see Figure 1.56).

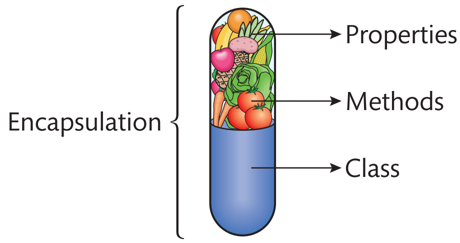


Figure 1.56: Encapsulation

The programmer creates specific interactions between different objects in order to solve the problem. As each class exists separately, they can easily be modified or adapted to reflect changes happening in the real world without having a negative effect on the whole solution. This makes OO languages very attractive to developers when they consider the demands of tackling ongoing maintenance.

You need to be able to recognise the structure of an OO program, identify its major components by name and state their purpose.

Figure 1.57 details the most common components that you will see in an OO program by re-implementing the stack simulator using C++.

```
#include <iostream>

using namespace std;

#include <stdio.h>

//constants
const int MAXSIZE = 5;

//function prototypes
char showMenu(void);
void clearBuffer(void);

class stack
{
 //hidden data (private properties)
 private:
 int data[MAXSIZE];
 int stackTop;

 public:
 //class methods
 stack() {stackTop =-1;}     //inline constructor

 bool stackFull(void);
 bool stackEmpty(void);

 void pushData(void);
 void popData(void);
 void showStack(void);

};
```

Constant

Encapsulation – a class which contains data (properties) and functions (methods)

Hidden data (properties)

Methods (functions)

```
//helper method to check if the stack is empty?
bool stack::stackEmpty(void)                          Method
{
  if (stackTop==-1)
  {
    return true;                                 Branches
  }
  else
  {
    return false;
  }
}

//helper method to check if the stack is full?
bool stack::stackFull(void)                           Method
{
  if (stackTop==(MAXSIZE-1))
  {
    return true;
  }
  else                                                Branches
  {
    return false;
  }
}

//method to push data onto the stack
void stack::pushData(void)                            Method
{
  int newData;
  cout << “Enter new Data “ << endl;
  cin >> newData;
  stackTop++;
  data[stackTop] = newData;
}

//method to pop data from the stack
void stack::popData(void)                             Method
{
  int poppedData;
  poppedData = data[stackTop];
  cout << “Popped data is “ << poppedData << endl;
  stackTop--;
}

//method to display the stack
void stack::showStack(void)                           Method
{
  cout << “Stack Top” << endl;
  for (int counter=stackTop; counter>=0;counter--)         Iteration
  {
    cout << counter << “ is “ << data[counter] << endl;
  }
}

//main program starts here                   Instantiation – creating an
int main()                                       object from a class
{
  //local data
  char menu;
  stack mystack;
```

```
  //main loop
  do
  {
    menu = showMenu();
    switch (menu)
    {
      //push chosen
      case ‘1’ : if (!mystack.stackFull())
            {
              mystack.pushData();
            }
            else
            {
              cout << “Sorry, stack is full.” << endl;
            }
            clearBuffer();
            break;

      //pop chosen
      case ‘2’ : if (!mystack.stackEmpty())
            {
              mystack.popData();
            }
            else
            {
              cout << “Sorry, stack is empty.” << endl;
            }
            clearBuffer();
            break;

      //show stack chosen
      case ‘3’ : if (!mystack.stackEmpty())
            {
              mystack.showStack();
            }
            else
            {
              cout << “Sorry, stack is empty.” << endl;
            }
            clearBuffer();
            break;
    }

  } while (menu != ‘X’ && menu !=’x’);

  cout << “Goodbye!” << endl;

  return 0;
}

//show the stack simulator menu and get option
char showMenu (void)
{
  char option;
```

Iteration

Function call

Branches

Calling object's methods

Function call

Calling object's methods

Function call

Calling object's methods

Function call

```
  cout << "\n\nStack Simulator" << endl;
  cout << "1 - Push data" << endl;
  cout << "2 - Pop data" << endl;
  cout << "3 - Show stack" << endl;
  cout << "X - eXit" << endl;
  cout << "Enter choice: 1,2,3 or X:" << endl;
  cin >> option;
  return option;
}

//clear keyboard buffer of any stray return/line feeds
void clearBuffer(void)
{
  while (getchar() != '\n');
}
```

Input and Output statements

Iteration

▶ **Figure 1.57:** C++ program using many common OO features

Key term

Polymorphism – this comes from the Greek word, meaning 'many forms' and is a core feature of the object-oriented (OO) programming paradigm. In C++, common uses of polymorphism include the 'overloading' of functions and operators, thus changing their use (but not their name) depending on the context in which they are used.

Although this example includes many OO features, some are not required in this solution. These include the following.

- **Inheritance:** This is the ability to create derived (or child) classes from a base class in order to reuse existing code (saving development time) and help to extend existing solutions with new functionality.
- **Polymorphism:** This involves using OO concepts, such as overloading, where multiple methods with the same function name exist but each accepts arguments of different data types or a different quantity of arguments. This allows the programmer to use just one named function to perform a set task but alter its behaviour depending on its use.

Again, at this level of study, you should be able to interpret programs of this complexity (or greater), be comfortable with the challenge of identifying the different components of OO programming and recognise how they connect to form a fully working solution. The key to OO programming is the potential reuse and extensibility of the program code to create amended and new solutions.

Event-driven programming

This is a popular paradigm for the development of graphical applications which respond to events generated either by the system (for example a system clock) or the user (for example a mouse click).

Event-driven programs typically work non-sequentially, with users able to select the operations they want to perform rather than follow the preset inputs of a more rigid program structure.

Developers focus on programming event handlers, which is the code that specifies the actions to perform when a particular event is triggered via a listener. A listener is a process that waits for a certain event to happen. For example, if a File->Open menu option is clicked, a file open dialog will be displayed.

Common event-driven programming languages include:

- Microsoft® Visual Basic .Net
- Microsoft® Visual C++
- Microsoft® Visual C#
- Oracle® JavaScript

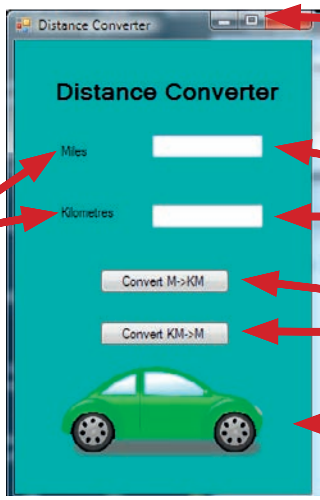
You may notice that some programming languages, such as Microsoft® Visual C++ and C#, offer features also common in object-oriented (OO) programming paradigms; this is not unusual – many modern programming languages are actually hybrids of multiple paradigms.

Event-driven programs work by operating in a **main loop**. This loop is effectively **listening** for different **events** to occur. Some events are **user generated**, for example clicking a button, and others are **system generated**, for example low RAM.

When the event is **triggered**, a suitable **event handler** (a specially written sub-routine called a **callback** is executed which responds to the event. The following simple example is created using Microsoft's® Visual Basic .Net. It uses a form, as shown in Figure 1.58, to provide a user interface and event handlers to process click events that are triggered from the form's buttons. Figure 1.58 also shows the Microsoft® Visual Basic .Net program code which handles button click events.

Link

Event handling is a core aspect of creating mobile applications, as shown in *Unit 17: Mobile Apps Development*.




```
Public Class Form1

  ‘constant (some rounding errors)
  Const MtoKM As Decimal = 1.61
  Const KMtoM As Decimal = 0.62

  ‘variables
  Dim miles As Decimal
  Dim kilometres As Decimal

  ‘event handler M->KM button
  Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) \
  Handles Button1.Click

    miles = Val(TextBox1.Text)
    kilometres = miles * MtoKM
    TextBox2.Text = Str(kilometres)
  End Sub

  ‘event handler KM->M button
  Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) \
  Handles Button2.Click

    kilometres = Val(TextBox2.Text)
    miles = kilometres * KMtoM
    TextBox1.Text = Str(miles)
  End Sub

End Class
```

Constants

Variables

Event handlers

Calculations forming the core actions of the event handler

Figure 1.58: Microsoft® Visual Basic .Net form and program code which handles button click events

Although this example includes many event-driven features, some are not required in this solution. These include **service oriented processing** and **time driven processing**.

Again, at this level of study, you should be able to interpret programs of this complexity (or greater), be comfortable with the challenge of identifying the different components of an event-driven programming and recognise how they connect to form a fully working solution.

The key to event-driven programming is realising that, unlike creating a procedural solution that leads the user through a sequence of actions, an event-driven program simply responds to different events that the user or computer may trigger.

Key terms

Service oriented processing – this is the breaking down of complex problems into a collection of separate (but potentially linked) processes, each providing a specific service for client applications.

Time driven processing – this is a form of event-driven programming whereby each process is triggered by a time-based event, typically using a computer's real-time clock (RTC).

Coding for the web

In this section, you will learn how to interpret, analyse and evaluate the use of code written for web languages.

The characteristics, features and implications of mark-up and web languages

Mark-up is a form of language used to specify the content (but not formatting) of a document in a structured manner using special tags. For example, in HTML, <p> tags are used to denote the start and end of a paragraph:

```
<p>A new paragraph</p>
```

Other mark-up languages can be used to represent complex data structures in a platform-neutral manner, such as entries from a CD library stored in XML, as shown in Figure 1.59.

```
<CD>
 <TITLE>Let's dance</TITLE>
 <ARTIST>Bowie, David</ARTIST>
 <YEAR>1983</YEAR>
</CD>
<CD>
 <TITLE>Rubber Soul</TITLE>
 <ARTIST>The Beatles</ARTIST>
 <YEAR>1965</YEAR>
</CD>
```

Figure 1.59: XML data

XML is often used as the preferred format to transfer data between different computer systems and applications, such as when exporting and importing data between relational database systems that are normally incompatible.

Because HTML focuses on the content of a web page, modern website development uses a separate technology called cascading style sheets (CSS) to add formatting, for example to change the colour of text or specify a font size or horizontal alignment.

For example, if you add **inline CSS** to the previous HTML example, the CSS formatted paragraph would now appear like this:

```
<p style="font-family:arial;font-size:32px; color:red">A new paragraph</p>
```

the CSS formatted paragraph would now appear like this:

A new paragraph

Web technologies such as HTML5 and CSS3 (the latest standards) can be used to create wonderful web page designs with attractive layouts and engaging content.

It is the job of the application's web browser to render the HTML and CSS on screen for the user; programmers tend to call this 'static' content as each page has been created manually by a web developer.

After completing your studies, you should be capable of writing and interpreting HTML5 and CSS3 with relative ease.

Link

For more detailed information on mark-up and style sheet technologies, see *Unit 15: Website Development*.

Uses, applications and implications of mark-up and web languages

HTML and CSS render very quickly on modern devices and, because there is a recognised world-wide standard, most web browsers render these pages very similarly to give users a consistent experience.

Free web browsers are available for a multitude of computer platforms, from powerful PCs running Microsoft® Windows to mobile phones running Android. This means that HTML and CSS achieve a form of platform independence: with appropriate internet connectivity, any web page created in standard HTML and CSS can be viewed on any device – anywhere in the world. This fact alone makes web page development technologies exceptionally powerful.

In addition, although standard web pages use a protocol called HTTP (hypertext transfer protocol) to transfer data between a web server and a web browser client, a secure version called HTTPS (hypertext transfer protocol secure) can be used to communicate sensitive data, such as passwords and electronic payment information.

The development and refinement of web-based internet services has fundamentally changed the way in which people communicate socially, shop, work, learn, listen to music, watch television and even do their banking. HTML, CSS and other web languages are at the very heart of these changes.

Although HTML5 adds extra interactive possibilities to a web page, other web technologies have been providing dynamic content for many years. These scripting technologies are said to be either client side or server side.

Uses, applications and implications of client-side processing and scripting

Client-side processing is achieved through the use of event-aware scripting languages such as JavaScript and VBScript™. JavaScript, which may be part of a HTML page or stored in a separate .JS file, is freely viewable once downloaded. Structurally, it is similar to many languages in the C family, as shown in Figure 1.60.

```
<script>
  var message = “Hello”;
  var length = message.length;
  alert(“Your string “+ message + “ is “ + length + “ character(s) long”);
</script>
```

Figure 1.60: JavaScript pop-up message

These client-side scripting languages are executed from within the web browser and are used to add event-based interactivity to a web page. For example, they can:

- validate user's input into HTML data forms, e.g. postcodes and telephone numbers
- perform bespoke animations
- bring up pop-up message boxes
- provide search boxes

- perform spell checking
  - provide auto-completion of forms
  - enable real-time updates to a page, e.g. breaking news or stock ticker
  - provide partial page updates without reloading, using a technology called AJAX (asynchronous JavaScript and XML).

Although JavaScript can be used irresponsibly, generally this is not the case. However, because of this possibility, JavaScript support can be switched off in a web browser client's settings. If this is done, even accidentally, it is possible that a coded solution using the web as its platform could be disrupted.

Comprehensive JavaScript libraries such as jQuert and AngularJS are used commercially to provide powerful built-in functions that make client-side development more flexible and productive.

Uses, applications and implications of server-side processing and scripting

In contrast to client-side processing, there are various scripting languages that are executed on the web server and which generate HTML, CSS (and even JavaScript) to send to the web server client.

Popular server-side scripting technologies include:

- PHP (PHP Hypertext pre-processor)
- Microsoft® ASP .Net
- JavaScript Server Pages (JSP)
- Lua
- Python®
- Ruby
- Perl.

Because the server-side script code is executed on the server, it is never viewable on the client's web browser, making it much more secure for commercial use. Server-side scripting languages often have considerable interaction with the server's operating system and can perform many complex tasks including:

- encryption
- database connectivity, querying, manipulation and management
- email generation
- file creation and management
- user authentication
- HTML form processing.

Server-side scripting languages, such as PHP, are also structurally similar to the C family, as shown in Figure 1.61.

```
<?php
$message = "Hello";
$length = strlen($message);

print "<p>Your string $message is $length character(s) long</p>";
?>
```

Figure 1.61: PHP server-side script

Once this server-side code is executed on the web server and sent to the client, only the underlying HTML code remains:

```
p>Your string Hello is 5 character(s) long</p>
```

Server-side scripting is a very powerful tool and is at the heart of any online retail site, as its ability to process search requests, query stock and customer databases and generate HTML quickly and painlessly makes it an ideal instrument for implementing code solutions on a web platform. The only major disadvantage is that its translation and execution add extra processing load onto the web server.

Issues and implications of implementing code on a web platform

Implementing a code solution on a web platform has a number of advantages and disadvantages.

Advantages

- Code solution runs inside a web platform so may not have full access to the underlying operating system or hardware, making it secure.
- Easy to integrate media such as images, video and music.
- Easy to link to other resources, e.g. spreadsheets, documents, electronic slideshows.
- Code does not need to be installed.
- Code will always be the latest version as updated on the server.
- Usage of the code solution may be easier to monitor.
- Potentially good integration with other online services, e.g. user support, email, instant messaging and social networking.

Disadvantages

- The HTML, CSS and JavaScript is usually viewable in the web browser client, so is not secure and can be reverse-engineered and copied relatively easily.
- Requires a client web browser.
- Code running inside a web browser can be less efficient and slower than code running natively on an operating system.
- An internet connection is probably necessary.
- Security and access may become an issue.
- Requires hosting which is usually not free.
- A target for hackers and website defacement.
- Not all web browser clients support web technology standards equally well.
- Overall functionality can be limited when forced to work within a web browser environment.

Translation

Sometimes it is necessary to translate a coded solution into another programming language. This process is known as 'porting'. A practical example of this could be converting an older procedural C program into a new OO C++ solution.

Reasons for translating code from one language to another

Why translate code from one language to another?

- Programming language can no longer meet the changing needs of the solution.
- Support for the existing programming language is ending.

- Existing programming language has identified security flaws.
- Change of hardware precludes continued development with existing language.
- Change of business/industry/sector policy regarding preferred programming language.
- Modern programming trends, e.g. newer programming languages.
- Availability of developers with suitable experience.

Benefits of translating code from one language to another

What are the benefits of translating code from one language to another?

- New solution uses more modern programming language.
- New solution is more extensible, easier to update and expand.
- Development time is reduced and changes can be made more quickly.
- Costs are reduced.
- Developer support is improved.
- New language features assist creation of new functionality for the solution.

Drawbacks of translating code from one language to another

What are the drawbacks of translating code from one language to another?

- New solution may be inferior to the original; compromises affecting performance may be made as exact translation is not possible.
- Solution may not be as efficient.
- Solution may prove to be overly time consuming to port to new language.
- New programming language is overrated; although some improvements may be possible, other drawbacks with the new language may be discovered.
- Developers for new programming language may be scarce and therefore expensive to hire and/or train.
- Less community knowledge about new programming language to help with bugs.

The implications and impact of translating code

Changing the programming language which creates a solution will always have effects.

The implications and impact of doing this on users, organisations and developers will now be considered.

Users

- Changes their perception of the program's performance – may be better or worse.
- The program may no longer have the same functionality.
- Lack of bug-fixes and maintenance on the old solution while the new solution is being created.

Organisations

- Purchase of new development environments, new programming language and new support materials.
- Potential hiring of new programmers or training of existing programmers.
- Developing a new solution costs money so there will be a need to justify this.
- Review of the new programmed solution – was it worthwhile?

Developers

- Learning of new skills.
- Personal and professional challenge – learning and developing with a new programming language.

- Dealing with the expectations of the users and the organisations.
- Challenges when debugging a new programming language.
- Finding the most efficient solutions using a new programming language.

Alternative ways to implement current code base

If translating a solution to a new programming is not practicable, it may be necessary to consider other ways of implementing the current code base. This could include:

- exploring new versions of the programming language
- using improved third-party function libraries
- investigating faster compilers or interpreters to improve performance
- identifying performance bottlenecks in the solution and creating improved algorithms to make the code base more efficient
- asking newer developers to view the code to see if they can recommend improvements and fresh ideas
- Whatever the programming paradigm being used, the ability to debug programs, identify errors and confidently fix them remains absolutely essential.

Research

Many opportunities exist to research and explore different programming languages online through the use of virtual development environments that let the visitor select a particular language, key some basic program code and execute it. To access a typical website with this functionality go to:

http://www.tutorialspoint.com/codingground.htm

PAUSE POINT Can you explain what the assessment outcome was about? What elements did you find easiest?

Hint Close the book and think about the different programming paradigms and their key features.

Extend You have seen a number of popular programming paradigms; which approach do you think is the most prevalent in the software development industry and why do you think this is so?

Assessment practice 1.2

A04 A05

Read through the following scenario and, using a range of programming paradigms and different types of programming language, create solutions for the following problem.

A program is needed that generates a results table for a local youth football league. The league currently has 20 teams and they play each other, home and away, over the period of 8 months from August through to March.

In common with other leagues, a win is awarded 3 points and a draw is awarded 1 point. The top team is shown in gold and the next two in green. All other teams are shown in black, apart from the bottom three, which risk relegation; these are shown in red.

The program should track the number of games played, won, drawn, lost, goals for, goals against, goal difference and the current number of points.

The user should be able to update this table by selecting two teams using their current table positions and then entering the result of the match. Invalid table positions and nonsensical results, e.g. teams scoring negative or very large quantities of goals should not be allowed. In addition, teams should only be able to play each other twice per season (home and away).

Once this is completed, the table should be automatically updated and displayed in an attractive manner.

Plan

- Do I fully understand the nature of the problem?
- Am I clear about what I am being asked to do?
- Do I know how to handle the data needed in this program?
- Do I know how to structure the data needed in this program?
- Can I identify which operators, built-in functions, validation methods, control structures and algorithms may be required to solve this problem?
- Am I able to create versions of this solution using different types of programming language and compare their effectiveness and suitability?

Do

- I know what it is that I am doing and what I want to achieve.
- I can use a variety of programming paradigms to create a working program.
- I can differentiate different types of programming language and can make effective judgements about their suitability when implementing a solution.
- I start from the beginning of the scenario and work through to the end.

Review

- I can explain different programming paradigms and their importance to the solution of a problem.
- I am able to create solutions using a variety of programming languages.
- I am able to analyse and evaluate their characteristics, uses, issues and implications.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

Further reading and resources

Books

Johansen A – *Python®, The Ultimate Beginner's Guide* (CreateSpace Independent Publishing Platform, 2016)

Flanagan D – *JavaScript: The Definitive Guide* (O'Reilly Media, 2011)

Nixon R – *Learning PHP, MySQL & JavaScript* (O'Reilly Media, 2014)

Schildt H – *C++: A Beginner's Guide, Second Edition* (McGraw-Hill, 2003)

Schildt H – *C# 4.0 The Complete Reference* (McGraw Hill, 2010)

Moore K – *Karl Moore's Visual Basic .NET* (Apress, 2002)

Websites

Coding Ground – Tutorials Point: **http://www.tutorialspoint.com/codingground.htm**

Microsoft® API and Reference Catalog: **https://msdn.microsoft.com/library**

World Wide Web Consortium: **https://www.w3.org/**

THINK FUTURE

Dan Hardy

Junior Programmer in a software development team

I've worked in the team for just over a year. In that time, I've worked on various programs and I've already learned so much. Some of the programming tasks we tackle are very complex – I've found that I can't just sit at my keyboard and start writing program code without thinking about the solution properly. When I first started, my manager encouraged me to draw flowcharts to break down difficult business logic into simple steps. I find this approach really works, and I end up returning to my flowcharts a lot while I'm working on the next part of the problem.

When I started, I only knew Microsoft® C#. I was worried when I was asked to develop in other languages like JavaScript and PHP. After playing around with these languages, though, I discovered they had similar syntax to C#, and many of the C# concepts I knew were easy to translate. I guess you could say I started to identify the repeating patterns!

When I have to describe to my friends what programming is like, I tell them that it's mainly about problem solving. I spend a lot of my time refining complex problems into simpler ones and trying to spot any patterns in the problems I deal with.

Focusing your skills

Being adaptable

There are many different programming languages used in computing and the industry is constantly evolving. It is really important that you keep your skills and knowledge as up to date as possible. To get a position as a junior developer in a software development team, you have to be able to adapt yourself and your skills to meet market needs and to keep up with current programming trends.

Here are some questions to ask yourself to help you to do this.

- Which programming languages are in demand right now?
- What kinds of skills are sought-after? Is there a particular interest in specific operating systems or specific types of application?
- Are there any other skills related to programming that are in demand, such as knowledge of web technologies or relational databases?
- What experience of and exposure to different programming languages is expected of a junior developer? What skills are mentioned in software development job descriptions and vacancies?
- Can I transfer my programming skills to new languages and different development environments?

Getting ready for assessment

This section has been written to help you do your best when you take the assessment test. Read through it carefully and ask your tutor if there is anything that you are still not sure about.

About the test

This unit is assessed under supervised conditions. The number of marks for the unit is 90. Pearson sets and marks the test.

The external assessment will last for two hours. During the supervised assessment period, learners will be assessed on their ability to apply their computational thinking skills to solve problems.

Sitting the test

Listen to and read carefully, any instructions you are given. Lots of marks are often lost through not reading questions properly and misunderstanding what the question is asking.

Arrive in good time so that you are not in a panic.

The test should be carried out under supervised conditions.

- You may use a calculator.
- Internet access is not permitted.
- You must not bring anything into the supervised environment or take anything out without your tutor's knowledge and approval.

Avoid any distractions from a mobile phone!

Make sure that you arrive in good time for each test session and check that you have everything you need for the test ahead of time.

Remember that you cannot lose marks for a wrong answer, but you cannot gain any marks for a blank space!

Ensure that you have checked all sides of the assessment task before starting.

Ensure that you leave yourself enough time at the end to check through your work. Proofread and correct any mistakes before handing in your work.

Command words typically used in assessment

Most questions contain command words. Understanding what these words mean will help you to understand what the question is asking you to do.

Command word	Definition – what it is asking you to do
Analyse	Examine in detail a scenario or problem to discover its meaning or essential features. Learners will need to break down the problem into its parts and show how they interrelate. There is no requirement for any conclusion.
Calculate	Apply some form of mathematical or computational process.
Complete	Complete a diagram or process. This can be applied to problems/solutions of varying complexity.
Demonstrate	Illustrate and explain how an identified computer system or process functions. This may take the form of a piece of extended writing, a diagram or a combination of the two.
Describe	Give an account of something or highlight a number of key features of a given topic. This may also be used in relation to the stages of a process.
Develop	Provide a solution to a problem, typically using an existing system or structure that must be improved or refined.
Discuss	Investigate a problem or scenario, showing reasoning or argument.
Draw	Represent understanding through the use of a diagram or flowchart.
Evaluate	Review and synthesise information to provide a supported judgement about the topic or problem. Typically, a conclusion will be required.
Explain	Make a series of linked points and/or justify or expand on an identified point.
Identify	Assess factual information, typically, when making use of given stimuli. Requires a single word or short sentence answer.
Produce	Provide a solution that applies established constructs to a given computing problem.
State, name, give	Assess factual information. Requires a single word or short sentence answer.
Write	Produce a solution, or a mechanism used as part of a solution, to a given computing problem.

Writing long answers

Work out which questions you need to answer and then organise your time, based on the marks available for each question. Set yourself a timetable for working through the examination and then stick to it – do not spend ages on a short 1–2 mark question and then find you only have a few minutes for a longer 7–8 mark question.

If you are writing a longer answer, try to plan out your answer before you start writing. Have a clear idea of the point your answer is making, and make sure that this comes across in everything you write, so that it is all focused on answering the question.

Try answering all the simpler questions first then come back to the harder questions. This should give your more time for the harder questions.

If you finish early, use the time to reread your answers and make any corrections – this could really help to make your answers even better.

Sample answers

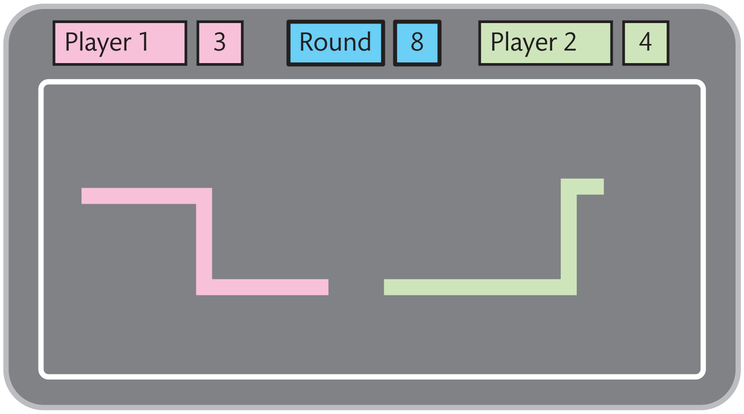
For some of the questions, you will be given some background information on which the questions are based.

Look at the sample questions which follow and our tips on how to answer these well.

Worked Example

Matteusz has been asked to create a simple 2D 'light cycle' game called 'Rainbow Line Warriors' which allows two players to race against each other and try to force the other to crash. Player 1 (red) starts on the left-hand side of the screen and is initially moving right. In contrast, player 2 (green) starts on the right-hand side of the screen and is initially moving left. Both players can move in four directions (up, down, left and right) using a different set of keys to control their cycle. The game awards 1 point for each victory, with the winner being the first to reach an agreed number of rounds.

A design for the starting screen is shown below.



Question 1

Identify three aspects of the game that would be represented as variables.

1.

2.

3.

3 marks

Sample answer:

1. Player 1 score
2. Player 2 score
3. Number of rounds

These answers are correct. The program would need to store both users' scores and the round number as variables. In this example, the easy answers are identifiable from the visual design; this will not always be the case. For example, what are the other variables involved? We could also consider the following:

Player 1 X position

Player 1 Y position

Player 2 X position

Player 2 Y position

Player 1 direction (Up, Down, Left or Right)

Player 2 direction (Up, Down, Left or Right)

Any of these answers would also be worth one mark as they demonstrate that you can analyse a scenario and discover its essential features.

Question 2

Matteusz knows that the 'tip' of each light cycle will be stored using x and y coordinates.

Player 1 (red) will use the following keys to control their light cycle: W – up, D – right, S – down and A – left. Matteusz also knows that the screen's origin (0, 0) is located top left.

Matteusz has written the following pseudocode that describes the movement of player 1's red light cycle, depending on the keys being pressed. He has used a separate function called 'CHECKCOLLISION' to deal with crashes. Unfortunately, he has made some mistakes and has not written it very well.

Rewrite the pseudo code to correct the errors.

```
INPUT PLAYER1KEY
IF PLAYER1KEY = W THEN
PLAYER1X = PLAYER1X - 1
CALL CHECKCOLLISION
ELSE
IF PLAYER1KEY = D THEN
PLAYER1X = PLAYER1X + 1
ELSE
IF PLAYER1KEY = S THEN
PLAYER1X = PLAYER1Y + 1
CALL CHECKCOLLISION
ENDIF
ENDIF
ENDIF
```

5 marks

Sample answer

```
INPUT PLAYER1KEY

IF PLAYER1KEY = W THEN
   PLAYER1Y = PLAYER1Y - 1
   CALL CHECKCOLLISION
ELSE
   IF PLAYER1KEY = D THEN
    PLAYER1X = PLAYER1X + 1
   ELSE
     IF PLAYER1KEY = S THEN
      PLAYER1Y = PLAYER1Y + 1
      CALL CHECKCOLLISION
     ELSE
       IF PLAYER1KEY = A THEN
        PLAYER1X = PLAYER1X - 1
        CALL CHECKCOLLISION
       ENDIF
     ENDIF
   ENDIF
ENDIF
```

This answer has identified and fixed most of the issues in Matteusz's pseudocode.

- Indentation has been added to highlight the structure of the IF…ELSE branches.
- New code has been added to deal with the A (moving left) key press which was originally missing.
- Code has been corrected to adjust the *y* ordinate, not *x*, when pressing W and moving up.
- Code has been corrected to adjust the actions when pressing S (PLAYER1Y should be adjusted, not PLAYER1X).

However, key press 'D' (moving right) still does not call the CHECKCOLLISION function – this has still not been identified. An improved answer should try to identify and correct everything.

Question 3

Matteusz's game is written using procedural programming techniques.

Explain how the structure of an object-oriented language would be used to manage the data and functions in the program.

4 marks

Sample answer:

A light cycle could be encapsulated as a class.

Each light cycle class could contain:

- Data/Properties such as
  - *x* position
  - *y* position
  - colour
  - score
  - player's name

- Functions/Methods such as
  - checkCollision
  - moveLightCycle
  - crashLightCycle
  - updateScore

The answer is acceptable. It has correctly identified a potential class, sensible properties and potential methods. Each of these is likely to be awarded a mark.

However, it could be improved by stating that two objects would be instantiated from the light cycle class for each player: red and green.

Fundamentals of Computer Systems 2

Getting to know your unit

Assessment
You will be externally assessed by a written examination.

The fundamentals of computer systems should be known and understood by every computing professional. This knowledge of how and why computer components, and the data they use, perform in certain ways is a tool used in the many varied roles of the computing industry. Understanding how different parts of a computing system work together enables us to identify and solve problems. Computer programmers use their understanding of how computers operate to develop better software.

This unit will introduce you to the various systems in current use, including their component parts, how they run software apps and handle data, and the methods used to transmit data. It will also consider the impact of these on organisations and individuals.

How you will be assessed

This unit is assessed by a written examination set and marked by Pearson.

The supervised assessment period is a maximum of 1 hour 45 minutes. You will be assessed on your knowledge and understanding of how computer systems work, including the role of hardware and software, the way components of a system work together and how data in a system is used. The number of marks for the unit is 80.

Grade descriptors

To achieve a grade you are expected to demonstrate the following across the essential content of the unit. The principle of best fit will apply in awarding grades.

Level 3 Pass

You are able to apply knowledge and understanding of key computing concepts to a range of familiar vocational contexts.

You are able to use knowledge of computing to deconstruct problems within common situations and apply standard conventions to produce solutions with interpretation.

You can identify the impact of effective and ineffective computer systems and recommend ways in which a system can be developed and/or improved (using given structures and criteria).

Level 3 Distinction

You are able to analyse complex information, data and situations, within vocational contexts, in order to draw conclusions and make valid observations.

You are able to synthesise knowledge and understanding of computing to deconstruct problems, drawing on various sources of information to develop effective solutions with justification.

You are able to evaluate the effectiveness of computer systems to make justified recommendations on their development and future actions that can be taken.

Assessment criteria

AO1 Demonstrate knowledge and understanding of computing facts, terms, standards, concepts and processes

Command words: complete, draw, give, identify, name, state

Marks: ranges from 1 to 5 marks

AO2 Apply knowledge and understanding of computing facts, terms, standards, concepts and processes to real-life scenarios

Command words: calculate, complete, demonstrate, describe, draw, explain, produce

Marks: ranges from 1 to 5 marks

AO3 Select and use computing technologies and procedures to explore likely outcomes and find solutions to problems in context

Command words: calculate, demonstrate, develop, explain, produce

Marks: ranges from 1 to 6 marks

AO4 Analyse and evaluate data, information, technologies and procedures in order to recommend and justify solutions to computing problems

Command words: analyse, demonstrate, discuss, produce, write

Marks: ranges from 6 to 12 marks

AO5 Make connections between the application of technologies, procedures, outcomes and solutions to resolve computing problems

Command words: evaluate, produce, write

Marks: ranges from 6 to 12 marks

Getting started

There are many different types of computer systems in use today. What computing devices do you own or use? What are their best and worst features? How would you improve them? What will you eventually replace them with?

A Hardware and software

Every computer system comprises both hardware and software. Hardware is the components you can physical touch. Software is less tangible, and includes the operating system and programs that control the computing system as well as data stored on the system.

Computer hardware within a computer system

Computer hardware consists of anything in the computing system that is not software, that is, anything physical, including the computers, screens, keyboards, mice, printers, cabling and other mobile devices and servers.

This section will look at the types, features and implications of choosing particular hardware, including their internal components.

Types of computer system

There are a number of different types of computer system, which we will look at in this section.

Multi-functional devices

Multi-functional devices are those that perform multiple functions, such as printers that include a built-in scanner. The scanner is often on the top of the device so that it can also be used as a photocopier. Therefore this one device can print, photocopy and scan documents when attached to any computer connected to it over a network or through a cable. Such multi-functional devices often include fax circuits so they can send or receive images to or from another fax machine through a telephone line.

Mobile devices are also multi-functional devices because they often incorporate into the one device a phone, a camera and computer.

Personal computers

Personal computers (PCs) are widely used for work and by individuals at home. The work carried out on PCs includes anything useful to an organisation and normally requires the use of software programs such as word processors, spreadsheets, databases, programming environments or other software apps. PCs can be broadly divided into desktops and laptops. Desktop PCs are usually left in one place, either in an office or in a home office, and often have a cabled connection to a network. Laptops can be moved to different places within a wireless, WiFi network. Laptops are useful for individuals who want the flexibility to use their device anywhere in their home or while they are away from home.

Mobile devices

Mobile devices are multi-functional, as they not only consist of a computer, but often include a camera and, in the case of smartphones or smartwatches, they are also phones. Mobile devices are often small enough to fit into your hand. Tablets and smartphones are the most common mobile devices. There has been a lot of progress in the development of mobile devices and the latest products are almost as powerful as a PC and have good internet connection and a long battery life. The mobile devices that we now carry around in our pockets are more powerful than many PCs that were on sale just a few years ago!

Research

Ask your family and friends what mobile devices they have owned. Also find out in what year they used their first PC, when they first used printers, and how these devices and mobile devices have changed during their lifetimes.

Servers

Servers are computers that run a network operating system such as Microsoft® Windows® Server. They are used to control the network and the users logged into the system on workstations (which are also known as client

computers). Servers are available in a variety of sizes to meet the needs of the smallest and largest organisations. They are powerful computers that usually have lots of memory and disc space and often have duplicated internal components to make them more reliable.

The purpose, features and uses of internal components

Internal components are the parts inside a hardware computer device. In this section, you will look at the features and uses of internal components in the different types of computer system.

Multi-functional devices

Multi-functional devices have an internal component called a processor to control them. The processor is a specialised computer that connects to a touchscreen and the network via a WiFi or wired network adapter. The touchscreen allows users to carry out functions such as photocopying and scanning and to change internal settings such as the network connection.

A network connection can be cabled into a **RJ45** port on the multi-functional device. Many smaller multi-functional devices also have WiFi so that they can make a wireless connection.

Key term

RJ45 – this is the common standard for network ethernet cable plugs and the socket or port that the cable uses to connect networked devices together.

The scanner mechanism in multi-functional devices is usually flatbed, often with a sheet feeder so that multiple documents can be scanned together without the need to place each one individually on the device.

The print mechanism of multi-functional devices usually uses inkjet, laser or solid ink technology. Inkjet technology is popular for use in homes and small offices. Laser and solid ink technologies are a good choice for businesses or organisations where there is a high demand for printing. Larger multi-functional devices include features for double-sided printing and stapling.

Link

See Output devices, in 'The hardware used in computer systems' for more detail on these technologies.

Personal computers

Desktop and laptop PCs have similar internal components, although it is usually much easier to change or upgrade these inside a desktop PC as a laptop has most of its functionality built into the motherboard, which cannot be changed.

Motherboard

The motherboard is the main circuit board that connects all the components plugged into it, including the processor, memory (RAM), backing storage (also referred to as secondary storage) and external components such as the keyboard, mouse, monitor, printer and network.

Processor

The processor (central processing unit or CPU) is a large chip that plugs into the motherboard. This is the component that understands and actions the program code, by working out any calculations, **logical operations** or **relational comparisons**. The CPU has one or more control units to action the code and arithmetic and logic units (ALU) for calculations and logic.

Link

See 'The concepts of microarchitecture' for more detail on how the processor (CPU) works.

Key terms

Logical operations – these are carried out between binary numbers using AND, OR, NAND, NOR, XOR and NOT logical operators.

Relational comparisons – these are carried out between numbers or text using operators such as less than (<), greater than (>), equal to (=), less than or equal to (<=), greater than or equal to (>=) and not equal to (<>).

The CPU is a powerful chip that does a lot of work and so it gets very hot. There has to be a method of removing the heat, which is usually a heat sink and fan. The heat sink sits on top of the CPU and is made from metal with a large surface area to conduct heat away and the fan blows air over the heat sink to make the heat dispersal more effective. Laptop CPUs usually have heat pipes instead of a heat sink as these are flatter. Without a way of removing heat, a modern CPU would get so hot during operation that it would melt away in around a minute.

Memory

When programs and documents are opened from backing storage they are placed in the memory (random access memory, RAM). The RAM operates much faster than the backing storage, enabling you to have multiple programs and documents active at any one time. The CPU needs the speed of the RAM because backing storage is far too slow for the CPU to use directly. Anything in the RAM disappears when the computer is switched off, which is why you must save your documents regularly when open, so that they are copied to the backing storage.

Backing storage

Backing storage is needed to store programs and data when the computer is switched off. The size or capacity of the backing storage represents how much storage space there is inside the device and is measured in gigabytes (GB) or terabytes (TB). There are a thousand gigabytes in a terabyte (see Fig 2.1). There are two main technologies used for backing storage – hard disc drives (HDD) and solid state drives (SSD).

- HDD: HDD is the older technology which works by spinning a disc round inside the device. Data is stored on invisible magnetic tracks (rings) on the disc using read/write heads which move to the track then copy data from the track to RAM (read) or record data from RAM to the track (write). Data is copied from the track to the RAM when a program/document is opened. Data is recorded from the RAM to the track when an updated or new document is saved.

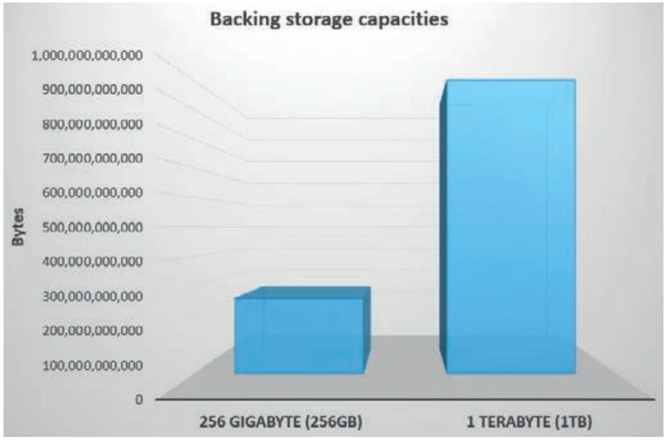


Figure: 2.1: Backing storage capacities

The performance of an HDD is influenced by the seek time, data transfer rate (DTR), spindle speed and cache size. These performance factors are explained in Table 2.1.

> **Key term**
>
> **Cache** – this is fast memory between a slow device and a quicker device that is used to prevent the quicker device from slowing down when the slower device is used. Processors contain cache, which is used for command queuing, to help program code run faster when commands are brought in from the slower RAM to the quicker CPU.

HDDs have RAM memory inside them that is used as cache to make the drive work more quickly by reading ahead or behind the current data so that the next data is already available (in cache) when needed. This technique allows speed matching of the slower to the quicker device. Write acceleration is also achieved as data can be quickly copied to cache then written to the slower disc.

The performance of an HDD can become poorer over time due to the read/write heads moving more between tracks as the data fragments. Fragmentation happens when files are deleted and the space is re-used for files which are of a different size and thus get saved to different parts of the disc. When this noticeably reduces performance the HDD needs to be defragmented (see Figure 2.2). SSDs do not need defragmentation as there are no read/write heads, so their performance is not slowed down by the need to move data to another part of the device.

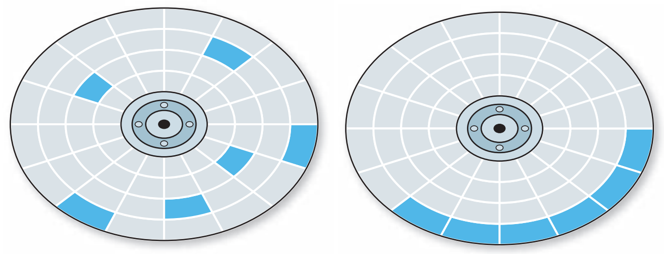


Figure 2.2: Before and after an HDD runs a 'defrag'

Table 2.1: HDD performance factors

Factor	Description	Units	Best performance
Seek time	The average time it takes the read/write heads to move to data in a track.	Milliseconds (ms)	Smaller seek time so finds the data quicker
Data transfer rate (DTR)	How quickly the data can travel to or from the HDD.	Megabytes per second (MBps) Gigabytes per second (GBps)	Larger DTR so transfers data faster
Spindle speed	How quickly the disc spins.	Revolutions per minute (rpm)	Larger spin speeds improve data transfer rates and seek times
Cache	Memory on the HDD to speed up data transfers.	Megabytes (MB)	Larger cache improves the speed on the HDD

- SSD: This newer technology uses electronic memory to store data, rather like the computer's RAM memory. SSDs use flash memory. This is non-volatile, which means it does not lose its content when power is removed (RAM memory on the other hand is volatile and loses its contents when power is switched off). Unlike the magnetic memory used in an HDD, an SSD has no moving parts, which has a number of benefits. The main benefit of using an SSD is that it is faster to access data since there is no wait for the disc to rotate or the head to move to the correct track as there is with an HDD. Data on SSDs can be directly read from any location and so they do not have a seek time, which gives them a much faster data access time than HDDs. This feature of SSDs means they do not suffer from the effects of data fragmentation either. With no moving parts, SSDs are shock proof and use less power than HDDs, making them more suitable for portable devices. The downside of SSDs is that they are more expensive than HDDs, and although prices have dropped since they were introduced, as of 2016 they are still around four times the price of a comparable HDD.

Both of these technologies use standard sizes of either 2.5 inches or 3.5 inches to allow easy fitting into PCs. The smaller size, 2.5 inches, is the usual size of drive used in laptops. The desktop connection between backing storage and the motherboard is usually through a **SATA** cable. The laptop connection is often the same, but the backing storage is plugged directly into the motherboard, rather than by means of a cable.

Key term

SATA – serial AT attachment (SATA) is the current standard for the cable connecting backing storage to the motherboard. This replaced the previous enhanced integrated drive electronics (EIDE) standard which was also known as ATA-2.

Discussion

With a peer or in a small group, discuss why an SSD is a better choice for a tablet than an HDD.

Mobile devices

Mobile devices differ from desktop computers in that size and battery life become important considerations. The main internal component of mobile devices such as smartphones is the **system on a chip** (SoC) which has the functionality of a motherboard and CPU. The RAM is usually separate to the SoC and **NAND flash memory** technology is used for the backing storage.

Key term

NAND flash memory – this kind of memory is used in SSDs, memory sticks and memory cards to hold documents, images and other files. This technology is non-volatile which means that it does not require power to store data. The name, NAND, is used to differentiate this technology from NOR flash memories. NAND and NOR are used to describe the algorithms used to store data rather than the type of logic gates used in their circuits.

System on a Chip – this is a single integrated circuit chip which combines all the functions of a computer. They are used in mobile systems because of their small size and low power consumption.

Other internal components inside mobile devices include the battery, long term evolution (LTE) modem, microphone, touchscreen display, sensors and camera.

Key term

Long term evolution (LTE) modem – mobile phones use long term evolution (LTE) modems to send and receive internet data.

Link

See 'The features and implications of embedded and mobile CPU architecture' for more about the internal components of mobile devices.

Servers

Server computers are often specialised versions of PCs with very large RAM, backing storage and powerful CPU(s). Many servers have redundant (duplicated) internal components so that, if one component fails, the redundant component can keep the system running. It is important that servers do not stop working, because they usually control and store data for many client computers across a network. Rack mounting is often used to keep a number of servers together in a cabinet.

Factors affecting the choice, use and performance of internal components

As a computing professional, you need to be able to recognise how the internal components of a computer system should be selected and the factors affecting this choice.

The performance of internal components is an important factor, but this has to be balanced with the user needs and cost. There is no point in paying for expensive and powerful hardware such as a gaming graphics card for a workstation that is only going to be used for basic office tasks.

Availability is another factor that should be considered. If you require a new computer system to be up and running now, in order that employees in a business or you at home can be productive, you will not want to wait for an internal component that is not expected to be on sale for another month. So internal components that are not available now would make poor choices for a computer needed this week. On the other hand, if you can wait, you might be able to get a higher specification component or the price of a specific component might come down.

Environmental concerns are increasingly important to consider as organisations recognise the need to 'do their bit' for the planet. Not only does considering environmental factors when choosing internal components help to improve the long term prospects of the planet, but can often save money and provide good marketing material for the manufacturer. Buying a PC with a more powerful CPU than is actually needed will use more power and cause it to get warmer than a CPU which is well matched to the intended workload, especially if this is from the most modern and up-to-date low-energy version of that processor. The reduction in electricity needed and air-conditioning costs for a large organisation using hundreds or thousands of these PCs could be significant.

The hardware used in computer systems

In this section, you will explore the hardware used in computer systems including input, output and storage devices.

Input devices

Input devices are anything used to control or get data into the computer system, so they may pass data to the computer but do not store data. The following are all examples of input devices that can be connected via a cable or wirelessly to a digital device.

- Keyboard – This can be connected to a computer, laptop, tablet or smart TV. Using the keys, users transmit instructions or input data into the digital device.
- Mouse – This is a pointing device that allows users to make selections on a computer screen.
- Webcam – This is a video camera that transmits moving images through an internet connection. Webcams make it possible to see friends and family on the other side of the world.
- Microphone – Used in conjunction with a webcam, a microphone allows us to transmit sounds across the internet or to give commands to the computer using our voices.
- Joystick – Similar to a computer mouse, a joystick allows users to make selections or navigate through a computer generated environment in a computer game.

Output devices

Output devices are used to get anything out of the computer system and provide information to the user from the computer. They perform a function using data or software stored in a computer. The following are all examples of output devices which can be connected via a cable or wirelessly to a digital device.

- Printers – Printers are used to reproduce text and images according to instructions they receive from any digital device. The printer receives data from the digital device, which it replicates on paper.
- Screens – A screen displays whatever software and/or data is being accessed on the digital device at any given time. They are crucial to human interaction with computers as, otherwise, we would not know what was stored on the digital device or be able to access the internet through it. A monitor is a peripheral device with a screen that is used in many computer systems.
- Projectors – These are similar to monitors as they are visual displays of whatever is being accessed on a digital device. Projectors are used to project the digital display onto a white or silver screen. They are mostly used to project films in cinemas (commercial and home) and for presentations in front of large groups of people (for example in schools or colleges) where there are too many people to gather around a small computer monitor.
- Speakers – Speakers convert digital audio signals into audible sound. They are used to listen to music or the soundtracks accompanying videos and enable users to hear people in other locations during web conferences.

Input/ouput devices

Some devices are both input and output devices.

- USB – this interface is used for a variety of purposes, including input devices such as mouse and keyboard, output devices like printers and storage devices like flash drives and external hard discs. USB has a number of versions, the current version being version 3. All the versions share the same physical connector and are backwards compatible, but the versions have progressively faster data transfer speeds with USB 3 being capable of up to 5 Gbit/s.
- Joysticks are generally input devices but you can get ones that have haptic feedback, which is a type of output. Haptic feedback is a tactile sensation that users feel in their hand in response to something happening in the software they are using. The most common use of haptic feedback is in computer games, as this enables users to feel sensations corresponding to things happening in the game, for example crashing into another car in a racing game.

Storage devices

Internal storage devices are components that are used to hold data and software for the normal running of the computer system.

- Hard disc drives – The HDD is an older technology, which is used where a lot of storage space is needed.
- Solid state drives – The SSD is the newer technology, which is used when the best performance is required (see the section on Backing storage for more detail).

External storage devices are peripheral devices used to hold data and software for the following reasons:

- to save space on your computer's hard drive
- to transfer files physically rather than over the internet
- to share files with another person or another digital device physically (if an internet connection is not available)
- to back up data in order to protect against damage or loss.

The following are examples of storage devices.

- External hard drives – These are used to back up data or to store large amounts of data. Digital devices such as laptops and tablets have limited storage space, so by storing data on an external hard drive you can increase your storage capacity.
- USB flash drives – These can store smaller amounts of data so that you can keep a copy of it or they can be used for temporary storage when moving data from one digital device to another.
- Server – This is a physical central location for storing and managing data between a number of networked computers.
- Cloud – This is a virtual server that is used for accessing and storing data via the internet.
- SD cards – SD (secure digital) cards provide high-capacity memory storage in a small space. They are most commonly used in small digital devices, particularly for storing still images.
- DVDs – DVDs (digital versatile discs) are a form of optical disc technology similar to CD-ROMs. They are most commonly used to store movies. Blu-ray discs are another form of optical storage designed for high definition video and have a higher capacity than DVDs.

Storage devices are often used as both input and output devices (known as I/O devices), meaning that you can copy or transfer the data from the device as well as onto it. Some storage devices will not allow you to reuse them, so you cannot change the data that is stored on them. These devices are read/write only.

Discussion

With a peer or in a small group, identify as many types of input, output and storage device as you can.

Link

To learn more about how the features of hardware can affect their performance see:

- Table 2.1: HDD performance factors, also sections on backing storage, the purpose, features and uses of internal components and computer hardware in a computer system
- Table 2.2 Display performance factors, in this section
- Table 2.3 Printer performance factors, in this section
- Table 2.5: CPU performance factors, in 'Cache', in 'The concepts of microarchitecture'.

How the features of hardware can affect its performance and the performance of a computer system

Most hardware components have features to improve the performance of both the component itself and the overall system. The overall performance of a system will be hampered by the slowest component part, so it is possible for one component to have an adverse effect on another. For example, a screen may not be able to display the high resolution produced by a high performance graphics card, so the performance of the graphics card would reduce down to the capabilities of the screen.

Display performance

Table 2.2 outlines the performance factors of display devices in computer systems.

Table 2.2: Display performance factors

Factor	Description	Units	Best performance
Refresh rate	How often the complete screen is displayed	Hertz (Hz). Most monitors have a 60Hz refresh rate.	A lower refresh rate will give you visible flicker and less smooth movement
Response time	A measure of how quickly each pixel of the display can change from one state to another (e.g. change of colour or brightness)	Milliseconds (ms)	Faster responses produce images with less trailing or ghosting which makes moving images seem blurred.
Brightness	The light output of a monitor	Candela per square metre, the larger the value, the brighter the monitor	The brighter the monitor the easier it will be to see in bright ambient lighting or sunlight.
Contrast	A measure of the difference between the light and dark areas of the image	A ratio, such as 600:1. The larger the first value, the higher contrast the monitor has	Higher contrast ratio will generally give you a higher quality image with the ability to distinguish between more colours

Printer performance

Table 2.3 outlines the performance factors of printers in computer systems.

Table 2.3: Printer performance factors

Factor	Description	Units	Best performance
Printing speed	How quickly pages can be printed	Pages per minute (ppm)	Larger number indicates a faster printer
First-out time	How long a delay before the first page starts printing	Seconds	Smaller number shows that the first page starts printing more quickly
Maximum printing resolution	How detailed the printed images can be	Dots per inch, often shown as across x down, e.g. 4800 x 1200	Larger numbers show a more detailed image
Monthly duty cycle	How many pages the printer can easily produce in a month	Pages	Larger number shows a more robust design

Factors affecting choice of hardware

There are many factors that determine our choice of hardware and these are discussed below.

User experience

User experience is the most important factor for many individuals and, to a lesser extent, for organisations. If the hardware does not give users a positive experience, then no one would choose it. Whether a user has a good user experience depends on whether the device meets their needs.

For example, an elderly person wanting to purchase their first mobile phone primarily for emergencies is likely to want a simple-to-use phone that makes and receives calls, with large keys, a clear screen and few, if any, other features. However, an individual looking to upgrade their smartphone will base their decisions on previous experience and may be looking for additional or enhanced features such as extended WiFi range, screen resolution and memory capacity. Other factors such as cost will also be a consideration for users and, while many sign up for monthly payment contracts, others may be wise to buy their phone on a pay-as-you-go contract.

There are number of aspects of user experience that will affect choice of hardware.

- **Ease of use** – Hardware that is easy to use or is more likely to save time is going to be preferred by most people, whether for personal or business use. Too many computer systems are overly complex and only a minor portion of those systems are used by most people.
- **Performance** – The level of desirable performance will vary depending on whether the technology is to be used for pleasure, business or academic research. The variety of functions that a device can perform is also a factor. For example, an elderly person may want a mobile phone for emergencies only so will not need particularly good performance and their choice will be based on a single criterion. However, a business may

base their choice on level of performance if it requires its employees to be contactable while on the move. Employees may also need to be able to access a central server for customer records, analyse financial data and produce complex reports. So businesses will make choices based on many criteria.

- **Availability** – Whether a hardware device is immediately available can influence our choices. We may make alternative selections if our first choice is out of stock. A business may buy on a large scale for multiple employees and cannot afford to wait for further supplies to become available. Many businesses require high availability server systems with redundant hardware so that single point failures do not bring the system down. Uninterruptable power supplies (UPS) may also be required so that power outages do not cause service loss.
- **Accessibility** – Accessibility or adaptability of technology may dictate hardware choices, especially if the user requires specialist devices such as touchscreens, ergonomic or light operated keyboards and mice, eye motion sensors, head motion trackers or sip-and-puff systems.

Research

Look at the different hardware and software systems than can be used by people with different disabilities.

User needs

Users will have different needs, which inform the criteria they use to make hardware choices. For example, your place of study requires computer systems that accommodate the needs of staff to use interactive whiteboards, printers and photocopiers. Computing students will use computers for their studies, and they are likely to need access to hardware such as printers and storage devices to save their assessments. Users also need to use their computers for different purposes. For example a user who needs a computer to do video editing will require a computer which has a powerful CPU, as this is a demanding application which also requires high capacity storage devices and a high performance graphics card. A different user who needs to run many web based applications at the same time might benefit from a PC with a multi-core CPU, lots of RAM memory and dual monitors, but may only require a hard disc of average size.

Specifications

The system **specification** will vary according to user needs. Software designers may require different types of system according to the tasks they are undertaking. For example, for programming, they might use fairly basic specification systems but they will probably want significant processing power and RAM, so they will require higher specification systems that provide these.

Key term

Specification – a description, often detailed, of the design of a hardware device. The specification gives details of the user interface that it uses, the processing power and amount of RAM, for example.

Compatibility

Compatibility means whether two or more things that need to connect and interact with one another do so properly. If they do not connect (physically in terms of cables or connectors) or interact with one another (work together), then they are incompatible. Decisions about choosing hardware for computer systems will be based on the needs of the whole organisation. Just as you might choose the same make for several devices out of personal preference and because you know they will work together, businesses are concerned about compatibility and whether the hardware components they choose will interact with each other correctly.

Cost

The cost of a computer system is very much a factor when making a choice about hardware. Businesses will allocate a budget for their computer systems. This budget is likely to include maintenance and upgrades. The cost varies according to the specifications and quality of manufacture of the hardware components. Cost is obviously an important factor for individuals and their budgets are usually much smaller than those of organisations.

Efficiency

The **efficiency** of a computer system must reflect user needs and is determined by the quality of the hardware components used. If a business is operating with low specification computer systems which under-perform, it will take employees longer to complete tasks and this will cost the business money. To achieve high efficiency, hardware needs to perform well with low power consumption.

Key term

Efficiency – a measurement of the useful output of a task or activity. A high efficiency means little wastage of time, materials or labour.

Implementation

Choosing hardware for a computing system can depend on the speed of its implementation. For example, individuals and organisations are likely to continue to use hardware which

operates in the same way as their current hardware because this avoids unnecessary loss of time for training. Hardware for a new computing system will be chosen for performance and perhaps to save space, increase mobility and reduce weight.

The urgency for implementing computing systems will influence hardware choices.

- Timescales – The preferred computing system (and its component pieces of hardware) may not be readily available or the lead time for implementation may be too disruptive for the organisation or individual. Suppose that your printer stops working and you want to upgrade to one with greater performance but cannot get the same make because it is out of stock. In order to avoid a break in your ability to print documents, as you have an assignment deadline coming up, you may choose an alternative make or model because it is available straight away.
- Testing – It is hard, and possibly irresponsible, to choose a computer system without having tested it first. Depending on what it is needed for, you might want to test it for comfort and weight, especially if you travel a lot for work. Someone who uses keyboards a lot will probably want to try out the keyboard for touch, size and positioning of keys.
- Migration to new system(s) – Transferring data, software and files to a new computing system can be stressful as well as time consuming. The process of migration needs to be straightforward and reliable. Some suppliers include data migration as a service when purchasing a new computing system, and will transfer contacts, favourites and apps, for example, to another mobile phone for you.

Productivity

Businesses very often make a decision to upgrade, change or implement new hardware to increase productivity. Productivity in industry is a measurement based on the quantity of work carried out or products made in a given time period.

Security

Computer system security is of constant concern. Frequent news reports feature examples of major breaches in security due to hacking into personal and business records. The hardware we choose to use needs to be secure and have up-to-date protection against both physical and cyberattacks. Organisations will make their choices about IT systems according to the security precautions in place, especially where sensitive information is managed, such as police, health and financial records.

Data storage and recovery systems

Data is stored in a computer system on an HDD or SSD. Either of these technologies can be used as components in redundant array of independent discs (RAID) or network attached storage (NAS) systems. RAID can be used in standalone computers and both RAID and NAS can be used in networked systems.

RAID

RAID systems are used when more than one HDD or SSD are used inside the same computer. There are several RAID standards that use data redundancy to duplicate data across the drives, so improving speed and data security. Table 2.4 provides more detail about RAID standards.

PAUSE POINT Can you identify hardware products that can be used to improve security for a computer system? How do these products enhance security?

Hint Identify methods that can be used to restrict access to a computer system.

Extend Find some biometric products that can read user fingerprints or eye retinas.

Table 2.4: RAID standards

RAID	Description	Tolerance	Number of drives
0	Block level striping only, to spread read/writes across the drives, making a fast system.	None	2+
1	**Mirroring** only to create identical copies of data on the other drive(s).	All but one drive can fail	2+
2	Bit level striping with **hamming code error correction** to enable rebuilding data with spread read/writes.	One drive failure	3+
3	Byte level striping with **parity error correction** to enable rebuilding data with spread read/writes.	One drive failure	3+
4	Block level striping with parity error correction to enable rebuilding data with spread read/writes.	One drive failure	3+
5	Block level striping with parity error correction to enable rebuilding data with spread read/writes.	One drive failure	3+
6	Block level striping with double parity error correction to enable rebuilding data with spread read/writes.	Two drive failures	4+

Key terms

RAID level striping – RAID can be used to spread saving and reading data over the drives to improve speed. Conceptually, this is thought of as making stripes from the data across the drives. Block level striping divides the data into 512KB blocks with each stored onto one of the drives. Byte level striping uses the same technique, but divides the data into bytes. Similarly, bit level striping divides the bytes into bits for storage.

Mirroring – RAID uses mirroring to save two copies of the data, so, if a drive fails, a good copy of the data can be brought back from the mirror.

Parity error correction – This is an error-checking technique by which an extra bit is placed at the end of each byte to make the number of 1s in the byte (and parity bit) an even number. This is called even parity; a similar technique where the parity bit makes an odd number of ones is called odd parity. Double parity error checking is when RAID writes a second set of parity across the drives as an extra safeguard against data loss.

Hamming code error correction – This is used in RAID systems to utilise parity bits to rebuild corrupted data.

NAS (Network Attached Storage)

NAS systems consist of a box containing one or more backing storage drives that can connect to a network as a simple file server. They can be used as an alternative to **cloud storage**.

Key term

Cloud storage – a virtual space for storing data provided through an internet connection. Advantages of this technology include being able to access the same data from almost anywhere, at any time and on any device.

Computer software within a computer system

Software is any program or application (app) that can run on a computer system. In this section, you will look at the purpose, features and factors affecting the choice of operating system, utility software and application software.

Operating systems

In this section, you will learn about the types of operating system, the role of the kernel, how the operating system manages networking and security and factors affecting the choice and use of user interfaces and operating systems.

Types of operating system

Every computer has an operating system, that is, software which is there to connect other software to the hardware components and to let the user control the computer system. There are various types of operating system used to control different types of computer system.

Real-time operating systems

Real-time operating systems (RTOS) are used in every interactive computer system and, as the name suggests, they process data in current (real) time. If a system is not real-time, it has a batch operating system to allow for long jobs such as updating records to be run automatically. Real-time operating systems (RTOS) are used in specialised systems where the computer must respond instantly (in real time) to inputs. RTOS are, for example, used in safety critical systems such as aircraft control and autopilot systems.

Single-user operating systems

Single-user operating systems are designed for computers used by one person at a time. These are usually multi-tasking so that the user can run several software

applications at the same time. However, some are single-task systems which can only run one user application at any one time, such as on a simple mobile phone (unlike a smartphone, which runs several software applications at the same time and is therefore multi-tasking).

Multi-user operating systems

Multi-user operating systems are designed for large computers that are used by many people at the same time for jobs requiring a lot of processing power, such as weather forecasting. They enable several users to operate the same software applications at the same time, for example through a server, and are usually multi-tasking so that users can run several software applications at the same time.

The kernel

The kernel is the central or core part of an operating system and is used to control and manage system components and tasks. The role of the kernel includes the control and management of the following components and tasks.

- Program execution: To load software, allocate resources, run and terminate when finished.
- Interrupts: For components such as the mouse to request servicing by the CPU.
- Modes: To switch between user modes, to run apps, and system (kernel) mode, to run low-level code that the operating system needs for operations such as reading memory.
- Memory management: Needed to allocate RAM to running programs and to reuse RAM when programs end.
- Multi-tasking: To allocate enough CPU and peripheral time to every running program so that they all work concurrently.
- Disc access: To read from or to write to backing storage.
- File systems: To control how data is stored and organised on backing storage. Operating systems use a variety of different file systems which use different addressing methods (to allow the operating system to save and retrieve files). Simple file systems such as FAT32 are used for low capacity devices such as USB flash drives where security is not required. More sophisticated files systems such as NTFS can address larger devices and provide additional features such as user access rights, improved ability to recover from disc errors and encryption.
- Device drivers: To allow the operating system to communicate with hardware components, such as a graphics card, to best use all their features and performance.

Networking and security

Modern operating systems are expected to have a role in managing networking and security.

Networking for a home PC will be peer-to-peer, involving no servers, with basic sharing of the internet, printing and some folders. Security on a peer-to-peer network is likely to consist of a power-on password (or user recognition) for each computer, a WPA2 password for joining the WiFi and passwords on shared folders.

Networking for an organisation will be client-server, with servers controlling the resources that user accounts can access. Security on a client-server network is based upon authentication of users when they log in, usually by typing in their ID and password, and then upon the parts of the network they have rights to access. Systems running on the network often have separate authentication log-ins.

Factors affecting the choice and use of user interfaces

There are a number of factors affecting the choice of user interface for an operating system.

Most people will be familiar with the graphical user interface (GUI) provided by Microsoft® Windows. Other common GUIs are those provided by Apple's® OS X® and Linux®. The GUI makes the operating system easy to use, with icons and menus that can be selected using a mouse. This type of user interface is very effective for multi-tasking with each program (task) in a different window. The performance speed or response times from a GUI will be slower than other user interfaces due to the processing power needed to produce the graphics compared with simple text-based command line interface (CLI) or menu interfaces, but the high level of usability of GUIs means that they are now the most popular kind of user interface on computer systems.

The CLI was very popular on operating systems such as DOS or CP/M before GUIs became widespread. Commands such as DIR, to show what is on a drive (the directory) need to be typed into a CLI making them a little more difficult to use than a GUI, as you need to remember the commands, although help is available to list and explain the commands. Every modern GUI includes a CLI for technicians to use.

Menu-based user interfaces make the operating system simple to use and well suited to single-tasking operating systems such as some mobile phones and smart TVs. This interface can also be available on a modern GUI-run operating system, where, for example, it will show the boot-up options if there is a problem with the installation.

Android®, Apple® iOS and Windows® operating systems use touchscreens, particularly for mobile devices such as tablets, smartphones and some laptops.

Factors affecting the choice of operating system

There are many factors affecting the choice of operating system (OS), many of which are discussed in this section.

- Type of operating system: Most people will want to have access to a single-user multi-tasking operating system so that they can carry out more than one function at the same time, and such digital devices are more powerful and useful than a single task based system. PCs used by individuals and businesses will be single-user multi-tasking. Some businesses may also have access to a multi-user operating system for collaborative working, which will be accessed via a server or the cloud.
- Origin of operating systems: Individuals and businesses can choose between proprietary operating systems, such as Windows® and Apple® OS X®, and open-source operating systems, such as Linux®. Proprietary operating systems are generally more widely used and support most popular software. However, open-source operating systems have the advantage that users can contribute to the features and capabilities of the operating system and they are generally less restrictive.

Link

For more about open-source as opposed to proprietary operating systems, see the 'The principles and implications of open-source operating systems and software'.

- Compatibility: Depending on the operating system used, there may be differences in file systems and issues of compatibility. Apple® computers use the OS X which has a file system based upon Unix. Windows® uses the NTFS file system by default, although the older FAT32 is still an option. OS X has compatibility issues when it needs to write to a NTFS drive as this needs a third-party utility.
- Intended use: If you are planning to become a software developer, data engineer or an IT technician who solves network and security issues, you will need to interact directly with the operating system by using the command line. For a relatively new digital user or someone who has little interest in anything more than basic functions that are simple to use, save time, enhance daily living or who simply wants entertainment, then a WIMP style GUI operating system is likely to be the appropriate choice.
- Scope of customisation: Open-source operating systems are far more customisable than proprietary ones. Therefore, if an individual or a business needs to be able to customise their operating system, for example if they are in the software development business themselves, then they are more likely to choose an open-source operating system. Home and business PC users who just want to use office software, browse the internet, send emails and play games, are better off with a proprietary operating system which is designed for ease of use and to support the relevant apps.
- Performance: The level of performance can be important in the choice of operating system . The fastest performing operating systems make modest demands on the hardware, so tend to be from the Unix family or older versions of Windows®. Performance also comes from being able to support the latest hardware and software apps, so the most current versions of the operating system are more appropriate as they run on the best currently available hardware.
- Cost: The cost of an operating system will be a factor for many individuals and businesses. As well as the initial outlay for purchasing the operating system, there may also be a cost attached to technical support from the manufacturer. Individuals may wish to pay for support directly from the manufacturer, while larger businesses will use their own IT technicians. When a new operating system is purchased, the individuals using it may well have training needs which will need to be paid for, particularly if the new operating system is significantly different from the one they were using previously. The cost of a new operating system will also be greater if it is not compatible with the software and documents previously used by a business, because these too would need to be updated.

Use and performance of operating systems

The fundamental use of an operating system is to provide the interface between users and the software applications on the computer system. The operating system is the link between the people and the software via the hardware (keyboard, mouse and screen).

Organisations usually have networks with client-based operating systems which are accessed from individual workstations. Individuals log on from their workstations to the server operating system.

Newer versions of operating systems usually have more functionality, which, if the hardware remains the same, means that the performance of the operating system slows down. Therefore many organisations upgrade their hardware at the same time as running a major operating system upgrade to ensure that their users are able to keep up or increase their productivity.

The amount of data you want your computer system to process at any one time will affect the performance of the operating system. A computer system running multiple applications and performing numerous operations

simultaneously, for example a computer system used to edit music and videos, will use considerable amounts of memory and processing power. In contrast, a computer system used entirely for administrative purposes, such as producing spreadsheets and word-processing documents, will use significantly less memory and processing power. In both examples, the operating system is single-user multi-tasking but the less memory required and the fewer applications running, the faster and more effectively an operating system can perform.

Utility software

Utility software runs alongside the operating system to carry out routine tasks to maintain and optimise the system operation. Utility software is there to add features and functionality to the operating system to help keep the computer running well. Some utilities are included with the operating system, such as defrag, while others can be purchased separately, such as anti-virus software.

The purpose, features and uses of utility software

The purpose of utility software is to undertake routine tasks. Routine tasks carried out by utility software include:

- security checks
- identifying and removing viruses
- cleaning up files
- software updates
- backing up data
- issuing warnings or alerts that updates or backups are required
- managing peripheral devices (such as cameras, printers and webcams).

Another use of utility software is to gather data to clean up the system and speed up performance by searching for junk data and purging it (removing it) to free up storage space. Utility software varies in the amount of memory it takes up according to its features.

Factors affecting the choice, use and performance of utility software

The choice of utility software is based on the needs of the system and whether a suitable utility is already present from the operating system installation.

Factors affecting the choice of utility software depend on what the user expects to get out of their device in terms of performance. If the device is mainly for browsing the internet or talking to friends using VOIP, it may not require software such as a backup utility. Users will choose utility software based on their needs. Most digital devices come with integrated software update processes, but it will be up to users whether they choose to add security and virus protection and to set up backup utility software.

Factors that can affect the performance of utility software include the amount of memory available on the digital device for utility software. Other factors are:

- bandwidth
- conflict with other utility software (such as multiple anti-virus applications running on one device)
- memory failures.

Application software

Application software (apps) enables users to interact with their computer systems to perform many different functions, including work and leisure activities. It is highly likely that you use several applications every day, such as Microsoft Word® and Excel®, as well as internet browsers or apps on mobile devices. Other examples of software applications are specialist business software used to manage stock control or accounting functions. They were all written by computer programmers. If you are studying computer programming, you may become familiar with programming software environments such as Windows® Visual Basic®, Logo and JavaScript.

The purpose, features and uses of application software

The purpose of application software is very wide ranging and includes software for work, entertainment, education, enterprise infrastructure, simulation, media development and product engineering.

The features of application software are focused on the expected usage. For example an app used to create presentations will have features such as transitions between slides when a slide show runs or a word processing app will have features to help document creation, such as different types of tab stops and control over margins.

Since the development of application software, business processes have been greatly enhanced. Application software enables businesses to cut costs, increase production and control quality through the use of automated digital processes and greater data processing power. There are also a number of businesses that benefit from the introduction of IT systems using application software that may not immediately occur to you. These include:

- farming – where software applications are used to manage harvesting, sorting, cleaning and packing produce
- textiles – software applications involved in the design and manufacturing processes in textile production.

Factors affecting the choice, use and performance of application software

For most organisations, the main factors affecting the choice of application software are compatibility with existing systems, cost and the user experience.

The usage of application software is largely determined by user needs. Performance is a minor factor in the choice of apps, as suitability is far more important and performance issues can usually be addressed by upgrading hardware.

One of the many reasons why Microsoft® products have been so popular worldwide is their versatility and usability (ease of use). Businesses needing general office applications mostly favour those which are familiar to the majority of users; in particular, Microsoft's Office® suite. When recruiting administrative staff, it is easiest to employ those with previously learned skills, such as how to use Microsoft Office® applications. Compatibility is also important. With so many people using Microsoft® Office, sending documents and spreadsheets using the Microsoft® Office file formats to other people is easy.

Apple Mac® products are most likely to be favoured by businesses in creative industries, such as advertising and graphic design, because the software that Apple® initially designed was more suitable for creative processes and, since then, they have built on that reputation. Microsoft® Office usually operates on PCs and does offer a version for Apple® Macs, although some Mac® users simply prefer the design of Apple® software.

The principles and implications of open-source operating systems and software

Operating systems and software are either open source or proprietary. Open-source operating systems are usually free, whereas proprietary operating systems are commercial and they cost money, either as a one-off payment or for a licence which can be renewed annually.

Behind all operating systems and application software there is programming code, known as the source code. You may have come across this if you have looked at the source code of a web page in **HTML**. There are two types of source code that are generally created – open source and proprietary.

Open source means that it is possible to edit and change the operating system (or application) software, as you have access to source code, whereas proprietary operating systems (and application software) are bought as-is, meaning they cannot be changed. An organisation with good in-house programmers would be able to modify an open-source operating system such as Linux®, to personalise it for that company. This means that not only can open source software be more closely customised to match an organisation's requirements, but it can also be much cheaper to implement. However, customising the software will require programmers who have the knowledge and skill to carry out the modifications and may also cause support issues since any problems that occur with the customised software will have to be dealt with by the organisation itself.

Key term

HTML – stands for hypertext markup language and is used to create web pages.

Link

For more on HTML, see *Unit 15: Website Development*.

For more about programming, see *Unit 1: Principles of Computer Science* and *Unit 4: Software Design and Development Project*.

Open source

Open-source code is a collaborative effort where more than one programmer contributes to the programming of the source code. It relies on peer review for testing its uses and functionality, and to fix any problems with it. Its main distinguishing feature is that the source code is open to use by anyone. Individuals can copy open-source operating systems and application software source code and, if they have the knowledge required, they can adapt it to their needs. Another distinguishing feature of open-source software is that it is usually free.

Link

For more about open source, see http://www.webopedia.com/DidYouKnow/Computer_Science/open_source.asp

One of the implications of using an open-source operating system, such as Linux®, and application software, such as OpenOffice®, is that it can be copied and modified by individuals with a malicious intent, such as expert programmers who do not share the same philosophy or responsibility to its integrity as the originators, or simply by inexperienced programmers. However, this risk can be mitigated by downloading from a reputable website.

On the positive side, open-source programming provides opportunities for individuals to be creative and share their knowledge and expertise globally and freely. It gives access to sophisticated software to people who would not be able to afford the proprietary software.

Proprietary

Proprietary operating systems and application software, unlike open source, have privately owned source code such as those produced by Adobe®, Apple®, Microsoft®, Pegasus® and Sage®. Proprietary OSs and apps will have been produced by teams of programmers who are employed by the company. Specifications and code of proprietary OSs and apps are kept secret within a company to avoid copies being made and to stop modifications being made by anyone other than the company's designated programmers. Proprietary operating systems and application software need to be licensed or purchased by individuals and businesses who can use the product, but cannot change the source code.

A major benefit of off-the-shelf proprietary software is that there is a lot of support available from other users, via the internet, training courses or books.

A further benefit of proprietary software for users is that the developers have control over the code so they are able to gather data on problems which occur and can use this information to correct programming issues. They are able to create code 'patches' that can then be delivered to registered users of the software.

One of the main implications of using proprietary software is that it can be costly, especially if the software is bespoke, that is, specially written for a particular client. Bespoke software often restricts further development to particular employed or contracted programmers.

Data processing

Data processing is the main use of computer systems for many organisations. Data can take many forms including customer records, stock information and financial accounts.

The use, features and implications of computer systems for data processing

You need to understand the ways in which computer systems are used for data processing and the implications of their use for both organisations and individuals.

Retail

Large supermarkets and other retailers use computer systems at the checkout to scan items sold and take payment, often registering points on a loyalty card. This gives the organisation useful information about stock levels and when to re-order goods and about the buying preferences of the customer. Customer information can be used to print vouchers that they can use on their next visit or to target mailshots and other sales promotions.

Supermarket systems consist of a large number of EPOS (Electronic Point of Sale) terminals at the checkouts connected to centralised server computers which record stock levels and automatically reorder items from suppliers when required.

Financial services

Banks have powerful computer systems to look after customer accounts and provide online banking and real-time information at cashpoints. These systems calculate interest and bank charges and interface with other systems to transfer money. Management use information from these systems to plan and market new products, such as savings accounts. Banks also have large distributed network systems with terminals located in branches connected to larger centralised server systems.

Farming

The farming industry is becoming increasingly automated. Livestock, such as chickens, are kept in barns that are automated to keep feeding and temperatures at optimum levels. Tractors use satellite-navigation systems to help control operations such as ploughing and seeding crops. These systems use sensors to detect environmental conditions such as temperature or geographical location and actuators to control equipment such as heaters or feeding machines. Usually the control software runs on an embedded system or laptop/desktop computer.

Manufacturing

Many manufacturing processes use computer controlled machines as they can perform tasks with a degree of accuracy and reliability that humans cannot match. Car manufacturing is heavily automated, with industrial robots used for tasks such as welding and paint spraying. Textile manufacturing uses automated looms and other machines to create cloth. Computer systems are used to program these systems and to control production according to the demands of the market. Computer controlled industrial machines generally use embedded systems or dedicated computers to control them. In a large manufacturing plant the control computers may be connected to central server systems to coordinate the manufacturing process

Engineering

The engineering industry uses sophisticated computer systems, based on high performance desktop computers running CAD (Computer Aided Design) software, to design and test components before manufacture. These designs can then be downloaded to computer-controlled machines to manufacture the physical product. Current advances in 3D printing make it possible to produce products anywhere without the need for shipping.

Science and medicine

Science and medicine are extensive users of computer systems, for example for exploration of computer models of theorems and for research into new drugs. Medical data can be used to identify the most effective treatments as well as poorly performing remedies. Doctors' surgeries typically use desktop systems connected to a small server to maintain patient records, while hospitals run a larger version of these database systems. Many medical instruments such as CT (computerised tomography) scanners use dedicated computer systems to create a detailed image of a person's internal organs from multiple X-ray images.

The role of hardware in collecting data

All data is entered into a computer system through hardware. Many hardware devices are specifically designed for data entry. These include keyboards, mice, touch screens and the various types of sensor.

The keyboard is still a very effective device for collecting data. Many data processing systems require paper-based information to be typed in so that the information becomes digital and suitable for further processing, such as sorting and arithmetic operations. Although a keyboard operator can make some typing errors, they can also add a lot of value to the activity as people can see problems in the source documents which an automated system might miss if the data was collected using a different method.

A mouse is not as versatile as the keyboard, but can still be effective at entering information, especially in completing on-screen forms using option buttons.

The touchscreen is a simple and effective data collection method that is often used in customer-facing systems such as bank cashpoints or automated check-in systems at GP surgeries.

There are many types of sensor used to collect data. One of the largest systems with sensors is the weather forecasting system used by the Met Office, which collects data from many thousands of sensors that measure rainfall, wind speed and direction, temperature, snow, atmospheric pressure, visibility, cloud and sunshine in the UK and across the globe.

Sensors that are used in computer-controlled environments could be as small as the central heating system in a small apartment or as large as a complex industrial process such as that used for refining oil. In both cases, the sensors collect a type of data that is then passed into the controlling computer system for processing. The output from these usually goes to relays which can switch electricity on or off to control hardware such as heaters.

Card readers are used in retail checkout systems to accept credit or debit cards to make a payment. The data from these transactions is sent to the card provider to record the sale. A loyalty card could be used to collect data for the retailer.

Many secure premises use swipe cards to control door entry. The data collected here records who used the door and when.

The role of software in collecting data

Software controls hardware in a computer system. Keyboards, mice and touch screens are likely to be the front end for software controlling a database when they are used to collect data. The data collection software, a spreadsheet or database, is used to collect, store and analyse the data. Sensors are likely to be part of a specialised system where software uses the data for a specific purpose such as creating a weather prediction model or controlling the temperature in an environment. Software can be used to indicate a problem with an input, for example showing a red light if a card swipe is not recognised.

Data processing functions

In this section, you will look at the main features of data processing.

Aggregation

Aggregation is a type of analysis where a large amount of data is processed into one or more summaries, which can then be used to create statistics or to provide data for another system. Data on potential customers could be packaged into aggregate reports which are then sold to businesses wanting to send out mailshots or run telephone-based marketing campaigns.

Analysis

Analysis is when some sort of meaning is extracted from data. There are many ways of analysing data such as exception reporting to identify poorly performing items in need of further attention. A supermarket could analyse sales of a new item and respond with a more prominent display or by withdrawing the item if sales do not meet expectations.

Conversion

Data processing can convert data from one form to another. A new system brought into an organisation may require a one-off conversion of data from the old system so that it can be used by the new, for example a payroll system. Conversions are often used in organisations to bring data held on a large central system, such as a selection of customer accounts, into a form that can be easily manipulated by the user, such as a spreadsheet.

Reporting

Reporting follows on from many data processing functions so that managers can understand the current state of their data or be made aware of any issues that have arisen during its processing, for example customers reaching their credit limit.

Sorting

Sorting is a basic method of data processing that is used to sequence data into **alphabetical or numerical order**. Sorting can involve multiple levels with the first level grouping identical items and the next level sorting items within each group.

Key terms

Alphabetical order – when data is placed in a sequence according to the order of letters in the alphabet, which means that anything starting with the letter B would be sorted into place after anything starting with the letter A and before anything starting with the letter C. Using alphabetical order, number 12 would be ordered after 1 and before 2.

Numerical order – when data is placed in a number sequence, usually ascending order, which means that 12 would be sorted into place after 11.

Research

What kind of systems are used for data collection and processing? What system does your school or college use? Do you have any family or friends who work in a large organisation? What kind of system do they use? What data is input and output? How is it processed?

Validation

Validation is used when data is entered to reject anything that is obviously wrong. Invalid data will not be allowed through validation. For example a product code could be two letters followed by four numbers so validation should reject anything which is not six characters in length and/or does not follow the pattern of two letters followed by four numbers.

The impact on individuals and organisations of using and storing data across multiple computer systems

In an ideal world, an organisation would be able to run all its various devices such as tablets, smartphones, laptops, desktops and servers integrated together into a single computer system. There have been real advances towards this goal in recent years, but using multiple computer systems are still the reality for many organisations. This section looks at some of the implications for individuals and organisations of using and storing data across multiple computer systems.

Access

Access to data is vital for staff who need to be able to work with customer accounts or other company information. This is easy on a standalone or networked computer but it can be difficult to synchronise data when multiple computer systems are in place which do not directly talk to each other.

Within a network, there might be an app that enables data from other apps to be updated overnight, so some data that is retrieved before the latest update could be a little out of date.

Discussion

With a peer or in a small group, discuss the benefits to businesses of synchronising data between mobile devices and an office computer network.

Cost

There could be a cost implication of storing data across multiple computer systems, due to the extra time needed to obtain information or from mistakes resulting from using out of date data on non-synchronised computer systems.

Implementation

Using data from another computer system could be via an update through a communication link or batch process. Sometimes the only way to use data from another system is to type it in using manual data entry, which is a time-consuming method that might result in human error.

PAUSE POINT What benefits are there to multi-level sorting of data? What real-life example can you find for data sorted in this way?

Hint Google 'multi-level sort Excel®' to find some examples of data sets that show ways of sorting.

Extend Why are the different sort orders useful for the data examples you found?

Tablet computers may be used by staff when visiting clients, with any new data synchronised through a WiFi network connection when they return to the office. Version control may be needed to ensure the best/latest version of data is always being used.

Productivity

Productivity will be reduced if data is stored across multiple computer systems, due to the time consumed in synchronising data, especially if this has to be done manually.

Security

Security might be improved with sensitive data on a very secure computer system, but, generally, using and storing data across multiple computer systems is less secure because there is more opportunity for hackers to access one of the systems. Security would be better in an integrated computer system as security can be controlled from one place and issues associated with securely transferring data between devices can be avoided. Storing data on mobile devices such as laptops or tablets may also create security issues, as these types of devices are more easily lost or stolen. Transferring data over an unsecured WiFi network to a mobile device being used out of the office might also present a security risk.

Backup and data recovery procedures

Backing up means to replace any data lost, for example, due to corruption or human error. The data is stored elsewhere (backed up) so that, if needed, it can replace the lost data. Backing up data is easier if it is all on a central server, as it can then be copied to the backup media in a single operation.

Backup procedures should be carried out on a regular basis, the frequency of which will depend upon the system requirements. A full backup might be made every week with incremental backups two or more times a day. A data recovery exercise should be carried out regularly to check that the data can be successfully restored to a system that has never been used for that data. This is important in case of a physical disaster, such as a fire or flood, affecting the location of the central computer server of an organisation.

Recovery procedures are also a lot simpler with one central server rather than lots of separate servers or networked computer systems.

Link

To learn more about backup and data recovery procedures, see the sections on 'Backing up data and IT disaster recovery plans', in 'Physical security' in *Unit 7: IT Systems Security and Encryption*.

Assessment practice 2.1

A02 **A04** **A05**

A national chain of stores has an established IT system which uses barcodes on products to check levels of stock on shelves in the shop, storeroom and on the checkouts.

Management are very interested in the performance of the stores, so they use MIS to bring this information to them.

- Describe the hardware and software involved in this system.
- What data processing functions are likely to be involved in transforming the store stock and sales performance into useful management information?

The national chain of stores is considering mounting a scanner and tablet onto each of a number of trolleys to trial customer self-service.

- What would be the impact of this service?

Plan

- Do I have any existing knowledge of the task?
- How much time do I have to complete the task? How am I going to successfully plan my time and keep track of my progress?
- What aspects of the task do I think will take the most/least time? How will I balance these?

Do

- I can make connections between what I am reading/researching and the task, and can identify the important information.
- I am recording my own observations and thoughts.
- I am recording any problems I am experiencing and looking for ways/solutions to clarify queries.

Review

- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).
- I can describe what strategies I employed to cope with failure and improve self-esteem.
- I can draw links between this learning and prior learning.

B Computer architecture

Computer architecture is the design of how components are connected and work together in a computer system. You need to understand the implications of the current computer architecture models and the impact of relationships between component parts in these systems.

Approaches to computer architecture

In this section, you will learn about the different approaches to computer architecture, including the features and uses of different computer architecture models and factors affecting their choice. You will consider the implications of using different architecture models as well as emulation.

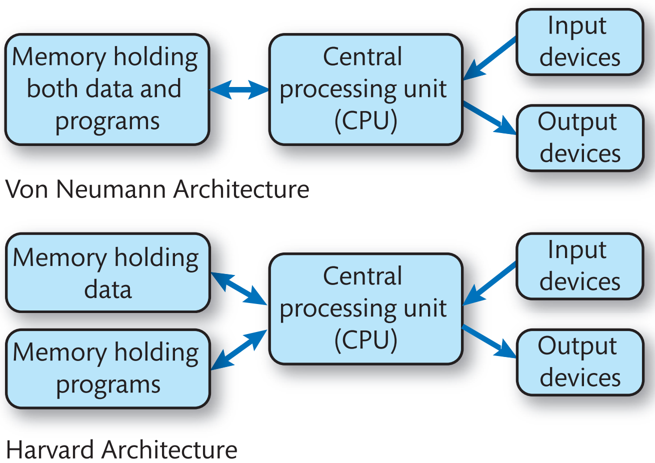


Figure 2.3: Computer Architectures

The features and characteristics of different computer architecture models

There are a number of different computer architecture models, each of which has its own features and characteristics.

Stored program model

The first computers did not use the stored program model, as programs were hard coded by setting switches. A technician needed to flip switches each time a new program was needed. Later systems used punched tape or cards to hold the programs and data. These have been superseded by modern computer systems.

The stored program model is normal in modern computers with the program(s) kept in electronic memory, which could be RAM or ROM. PCs, laptops and large systems use RAM, whereas ROM-based systems tend to be smaller and dedicated such as those used in a car engine management system.

Stored program model computer systems use either Von Neumann or Harvard architecture.

Von Neumann architecture

Most of the computers that you will have experience of using employ the Von Neumann architecture as they use the same RAM for both programs and data. The operating systems, apps and documents all share the same memory. Von Neumann architecture computers have a single processor that uses the **fetch-execute cycle** to run code from this memory.

Harvard architecture

The Harvard architecture, an alternative to Von Neumann, uses separate memory for programs and data. This architecture is used with embedded computer systems in devices such as personal video recorders (PVR) and **Arduino**.

Key terms

Fetch-execute cycle – this is the sequence repeated thousands of millions of times every second by a CPU to fetch the next instruction code from memory and then run that code.

Arduino – this is a small, cheap hardware device containing a CPU and input/output (I/O) ports, which can be used to explore programming and controlling devices.

Modern CPUs use Harvard architecture inside the chip to separate code and data into threads for faster processing, but the connection of the CPU to the motherboard and components will be Von Neumann.

Cluster computing

Supercomputers used by the Met Office to produce UK weather forecasting use the cluster computing architecture model. These computer systems have hundreds of thousands of processors and are capable of carrying out thousands of trillions of calculations every second. Simple home PC hardware can also be clustered together to provide a powerful system capable of much larger processing tasks than single machines.

PAUSE POINT Why do you think the Met Office needs so many parallel processors? Why would an ultra-quick single processor not be appropriate for this computer system?

Hint Find out how many input sensors from weather stations this system uses.

Extend What are the benefits to the Met Office of more processing power?

Uniform memory access and non-uniform memory access

Multi-processor computer systems, with more than one CPU, can share RAM between the processors. Uniform memory access (UMA) and non-uniform memory access (NUMA) are two techniques for sharing RAM between processors.

UMA is a technique for sharing RAM where each CPU can access any of the memory at the same speed. Individual CPUs can have their own cache memory allocated in the shared RAM. UMA systems share the same **memory bus**.

NUMA is an alternative technique for sharing RAM whereby each CPU can have some RAM allocated as local memory as well as sharing other (foreign) parts of the RAM with the other CPUs.

Key term

Memory bus – this is used to move data in a computer system between the CPU and the RAM.

Use and application of emulation

Processors are designed in families, for example the Intel CPUs found in modern computers have evolved from the 8086 processor from the late 1970s called the 'x86 family'. The 8088 version of this CPU was used in the original IBM® PC of 1981.

A family of processors uses the same **low-level program** commands (which can be expanded in later versions) and can usually run the same operating systems which provide the **API** needed for application software to run. An app will not usually run on a computer with an operating system from a different family.

Key terms

Low-level program – This is code written using instructions which map very closely to the CPU's actual instruction set. The CPU instructions are just binary numbers. In low level programming these are replaced by mnemonics. For example the CPU instruction for the add instruction might be 01000000 in a low level program this might be replaced by the mnemonic ADD.

API – this stands for application program interface, which is the place where code from an app can send calls (requests) to the operating system to make the hardware do something, such as opening a document.

Emulation occurs when software is run to provide an environment that is the same as that of a different processor family so that an operating system can run a program that was designed for a different processor. For example, a PS3 emulator could run on a Windows® PC so that the user can play a PS3 game on their PC. Emulation slows the computer system down as it adds an extra layer of software between the operating system and the hardware.

Link

For more about mnemonics and assemblers, see these key terms in 'The use and choice of instruction sets'.

Factors affecting the choice of different architecture models

The main factors affecting the choice of different architecture models are how suitable the architecture is and the cost of ownership.

Suitability of a computer architecture model is a complex mixture of the user experience, the apps currently available for the architecture and how well the platform integrates with the other systems run by an organisation.

Cost of ownership is calculated from the initial cost plus the ongoing expenses, which includes any licenses that require regular payment, and support costs.

Von Neumann with UMA is the default choice for a modern PC or laptop. This architecture performs well for a single user running one or more apps.

Harvard is the default choice for small dedicated devices as the code will be burnt into ROM with separate RAM memory for any changeable data that is entered.

Cluster computing with NUMA is a good choice where a massive amount of data from many sources needs to be processed quickly. The large number of CPUs can work in parallel to quickly process the data as it arrives.

UMA is a good choice for a computer with more than one CPU on the motherboard. Many networked server computers use multi-processor systems to provide good performance to the users accessing this system.

The impact of using different architecture models

The choice of architecture models for PCs and laptops is between the Apple® Mac family of hardware and the various manufacturers of PCs. The Apple® Mac family architecture has been developed during the last decade so that it is so similar to the PC that Windows® or Linux® can be installed onto a Mac® via the Boot Camp partition manager. It is, however, difficult to install OSX onto a PC as this operating system is locked down to Apple® hardware architecture, which limits hardware choice.

The impact from this choice is the operating system interface for the users, software and hardware compatibility. The Mac® OSX, Windows® and Linux® operating systems have different APIs, so need software apps written for these platforms. PCs have a wide variety of expansion hardware cards available, such as graphics, which are unlikely to work with Macs®.

Tablets and mobile phones have a close relationship between the hardware and the operating system, so can be classified as Android®, Apple® or Windows® devices. Android® is very popular for these devices with a wide range of apps available. The Apple iPhone® and iPad® families are also very popular, having a wide range of apps as well as offering good integration between these devices. Windows® phone devices have a much poorer choice of software apps but do offer the Windows® interface and some compatibility with Windows® apps.

The concepts of microarchitecture

Microarchitecture is the design of the very tiny circuits inside the CPU, motherboard and other computer components. Microarchitecture affects the choice of CPU for PCs and laptops and the choice of SoC for smartphones. In this section, you will explore the concepts of computer component microarchitecture.

Instruction cycles

An instruction cycle is when the CPU executes an instruction or command in low-level code. This is called a cycle because, when the execution is completed, the cycle restarts with the next instruction. Every CPU carries out many instruction cycles every second.

Execution speeds

The execution speed is the rate at which a CPU can process instructions. This section will look at the execution speeds in more detail.

Factors affecting execution speeds

The main factors affecting execution speeds are the clock speed and the design of the CPU. These and other factors are outlined here.

- Clock speed: The clock speed is how many instructions can be executed in a second. A typical processor can be rated at 4GHz, which means that it can execute 4,000 million instruction cycles every second.
- Design of the CPU: The design of the CPU circuits has a large impact on performance. For example, Intel designs include i3, i5, i7 processors. The performance of each of these processors at the same clock speed varies, with the i7 being the best and i3 being the slowest. The i7 has more design features compared with the i3, such as 8MB of cache (compared with 4MB) and turbo boost (so that the CPU can speed up for short periods).
- Cache memory: The cache memory in a CPU holds queues of instructions and data, ready for processing, to reduce the waiting time involved in fetching them from RAM and to help speed up writing to RAM. The greater the cache memory, the faster the execution speed.
- Number of processing cores: Multiple **cores** are able to execute code faster, for example a 4 core CPU can run four code threads at the same time, whereas a 2 core CPU can only run two at a time.
- Cooling: Cooling the CPU is important because if a processor gets too hot, it reduces the speed (throttles back) to generate less heat and so help cool itself down.
- Motherboard design: The design of the motherboard has a significant influence on execution speeds as this component connects the CPU to the RAM, video and backing storage.
- Amount of RAM: An inadequate amount of RAM can have a serious impact on execution speed as current code and data will need to be swapped out to backing storage. The trigger point for this **paging** depends upon the operating system and the apps and data open at any one time. For example, 4GB of RAM could easily be enough for one user but nowhere near enough for another user who needs several apps open at once or needs to work with large data sets.

Key terms

Core – the part of the CPU that is able to run code. Up until 2005, all processors had single cores as it was not possible to make the circuits small enough to fit two cores into one chip before that. 'Quad' (4) core processors are now common but processors with higher numbers of cores do exist.

Paging – this is a technique used by a PC or laptop to use the backing storage as an overflow area for RAM if the RAM fills up with programs and data, which would slow the system down. The technique is called paging because RAM is treated as a number of same-sized pages to make swapping in and out of disc easier.

Methods of increasing execution speed

The most effective method for increasing execution speed is to upgrade the CPU, motherboard, video system and RAM. A better CPU should increase the execution speed. The motherboard, video system and RAM move data in and out of the CPU, so a faster execution speed can be achieved through better components.

Some users like to over-clock their computer system. Over-clocking is when the system is set up to run the CPU at a faster speed that it has been rated for. There are risks in doing this because the faster speed will generate too much heat and may burn the CPU out or it may run unreliably, resulting in computer system crashes.

Implications of execution speeds

Faster execution speeds provide a better user experience as the system responds well and feels more responsive and satisfying to work with. A fast system encourages the user to explore new ideas and use more interesting software because they are not constantly waiting for the system to catch up with them.

Slow execution speeds can be quite frustrating to work with, because they involve waiting for simple tasks to complete such as the screen updating or loading documents.

More demanding software such as game development environments need fast execution speeds from the hardware used.

The use and choice of instruction sets

An instruction set is a range of low-level program code commands that are understood by the CPU. These instructions are called 'machine code' and are usually programmed using **mnemonics** through an **assembler**. Machine code is low-level code that has a direct influence on processing, which means that code can run very fast with the programmer being able to make the CPU do exactly what is required. The downside is that a lot of code is needed, as hundreds of machine code instructions are needed to run a program that would take a single high-level program command. Traditionally computers have complex instruction sets (CISC) with a large number of instructions, many of which are very rarely used in day-to-day programs. Reduced instruction set computers (RISC) only have those instructions which are commonly used, which makes for a simpler more efficient processor. Instructions sets also vary in the number of bits used for each instruction, with 8, 16, 32 and 64 bit instruction set computers in common use. In general the more bits used the larger the amount of memory that can be addressed by the instructions.

Key terms

Mnemonics – these are acronyms used in a low-level programming environment for producing machine code programs. Most machine code instructions are made from the opcode (command) and the operand (value or address the opcode is to use). Each mnemonic represents a single machine code opcode instruction, such as DEC which will decrement the operand (decrease by 1).

Assembler – this is a programming environment used for producing machine code programs. Use of low-level programming is quite rare as most code is written in high-level languages such as C#, which are a lot easier to use.

Pipelining

Pipelining is a technique used inside the CPU to help run code faster. Pipelining works by lining up code instructions inside the CPU so that several pipelines can run side by side at the same time. This is done to keep the CPU as busy as possible by processing the next code instruction as soon as the previous one is complete.

Pipelining breaks each instruction into four stages: fetch, decode, execute and write-back to RAM. As one instruction is being executed, the next is being decoded and the one after that is being fetched.

Cache

Cache is specialised RAM that helps to improve performance when a faster component needs to communicate with a slower part of the computer system. The cache is memory that is shared by both components with all data moving through the cache. Data sent from the faster component to cache is very quick, and the slower device can then take the data from cache at its own pace. This means that the faster component does not need to wait needlessly.

When the faster component needs data, more than is immediately wanted is brought from the slower device to cache so that, when the faster device needs more data, it is already in cache. For example, the CPU has cache memory inside the chip as this component is faster than the RAM (see Table 2.5).

PAUSE POINT Pipelining is a powerful technology that is built into modern processor CPU chips to help them run code more effectively. What is pipelining? How does it improve performance? What are the potential problems that need to be overcome in the design of pipelining inside these chips?

Hint Use the internet to research how pipelining works.

Extend Visit the Intel website to read their promotional materials for their current range of processors. What has Intel done to design their best use of pipelining?

Table 2.5: CPU performance factors

Factor	Description	Units	Best performance
Core design	Which family the CPU cores belong to	No units, based upon product names, e.g. i7	No performance figures
Number of cores	How many CPU core circuits are in the chip	Single, dual, quad, six, eight	A larger number of cores produces better performance
Clock speed	The number of processing cycles carried out every second	Gigahertz (GHz), number of 1,000,000,000 cycles in a second	A larger clock speed produces better performance
Manufacturing process	How tiny the circuits can be made	Nanometre (nm), 0.000000001 m	A smaller manufacturing process allows more circuits on the chip, which runs slightly faster as the parts are closer together
Cache	Fast memory in the CPU that holds instructions waiting to be processed	Megabyte (MB)	A larger cache gives a better performance as more code and data can be pre-loaded into the CPU

Registers

A register is a collection of circuits used to hold a byte of code in the CPU. When the CPU runs code, registers are used to hold the instructions and data and to help communicate with the RAM.

Link

For more about how registers are used, see 'Registers and register handling'.

Multi-processing and multi-threading

Being able to run different parts of the program code at the same time can obviously improve performance. Multi-processing and multi-threading are two ways of running different parts of the code at the same time.

- Multi-processing: This refers to hardware that is able to run more than one process at the same time. It could be several cores in the same CPU, several CPUs on the same motherboard or many CPUs in a larger system such as a supercomputer.
- Multi-threading: This is when the code in an app is run as several threads, each at the same time. For example, a modern word processor can have a thread to check for spelling and another to check grammar, while other threads respond to keyboard and mouse input. Multi-threading can also be applied to processor CPUs. If the current instruction cannot continue (perhaps due to waiting for data from the RAM) the CPU can switch execution to waiting processes which can continue.

The features and implications of embedded and mobile central processing unit (CPU) architecture

Embedded and mobile CPUs are used to control devices such as domestic appliances (washing machines, microwaves, etc.) and industrial machines such as a paint spraying robots, and to control systems such as a vehicle engine management system as well as smartphones and tablets. The architecture of the control for these systems is based upon a SoC, which means a system on a chip.

The SoC is a functioning computer system on a single chip. The CPU inside the chip is based upon circuits designed by manufacturers such as ARM, Zilog and Texas Instruments.

The connections between the SoC and the rest of the mobile device are a very important consideration for the designers of these devices who are looking to create easy control of specialised inputs and outputs through dedicated co-processors inside the SoC, such as 3D touch on a smartphone.

Most laptops use specialised mobile CPU architectures with processors that are designed to draw a small amount of power to maximise battery time between charges. Embedded CPUs used to control processes in machines require interfaces to sensors and actuators. For example an embedded system in a washing machine would need sensors to detect the water level and temperature and actuators to operate the water valve and heater.

The features and implications of microcomputer CPU architecture

A microcomputer is a computer system controlled by a microprocessor, which is a CPU chip. Although this is a broad definition that includes all laptops and many servers, microcomputer usually refers to a desktop PC system.

The main focus of microcomputer CPU architecture is the processor and motherboard, as these have the most impact on performance. The interfaces to backing storage, video and external devices can be bottlenecks which slow the system down, for example when you open files from backing storage.

The features and implications of server CPU architecture

For many businesses, server CPU architecture is a specialised adaptation of a microcomputer which can reliably control a network with many computer systems which need to be available 24/7.

The need for reliability can be met by incorporating redundant components. Redundancy is when a component in a computer system is duplicated. For example, a server can have two power supplies so that if one fails the other can take over. If a redundant component is a 'hot plug' it can be replaced without needing to turn the server off.

Servers also need a high enough speed and capacity to meet the needs of the many users connected to the computer system. Many servers have two CPUs on the motherboard with massive amounts of RAM and backing storage. RAID is common to maximise the speed of access to backing storage and to spread data across the drives, so a broken drive can be 'hot-plugged' for a replacement with no loss of data and without needing to turn the system off.

PAUSE POINT Many businesses use a server computer that is an enhanced version of a desktop PC. How is this different from a normal desktop PC? What features are likely to be present in a server computer?

Hint Close this book and brainstorm as many features that you can think of that would be useful in a server computer, for example lots of disc space.

Extend Visit the Dell UK website to research the features that they build into their more expensive servers. How does this list compare with your brainstormed list?

Registers and register handling

There are many registers inside the CPU of a computer system to hold the bytes of code and data when they have been fetched from the RAM for processing.

Types of register

There are a number of different types of register – general purpose registers and a variety of special registers – all of which you will look at in this section.

General purpose registers

General purpose registers are used in the CPU to hold data when code is executed. A low-level program could be written to loop through some data to count how many records match a search value, with one register being used to keep a count of how many matching records have been found and another register used to keep a count of how many iterations have been made.

Special registers

Special registers are inside the CPU and each is designed for a particular role during the processing of code.

- The accumulator is inside the ALU (arithmetic and logic unit) and is the part of the CPU where all the calculations and comparisons are made. This register usually holds one of the values to be used before the operation and always holds the result after the operation is complete.

- The instruction register (IR), also known as the current instruction register (CIR), is the place where each machine code instruction is held for decoding and execution by the control unit.
- When the CPU needs to read or write to memory, the address in RAM is placed in the memory address register (MAR) which is then sent by the address **bus** to the RAM so that the data bus can connect to that location.
- The memory data register (MDR), also known as the immediate access store, is inside the CPU at the end of the data bus. When the control unit sends a signal, data is copied from the MDR down the data bus to the RAM location specified by the MAR or from the RAM location to MDR.
- The role of the program counter is to hold the address of the next program instruction. Machine code instructions have different lengths, so when an instruction is fetched into the CPU, the program counter has the length of this code added to the address of the current instruction already held there, so it now points to the next. Before an instruction is fetched, the program counter is copied to the MAR so that the data bus can connect to that RAM location.

The function and purpose of general and special registers and their impact on the way in which computer systems perform

The function of the general registers is to be a resource for low-level programs and they fulfil the role of **variables** in **high-level programs**, allowing them to store data needed during processing.

The size of a register has an impact on the way in which a computer system performs. A 128 bit register is able to hold 16 bytes, while an 8 bit register can only hold 1 byte. Bigger registers need fewer clock cycles to process large calculations or to transfer data, because more bytes can be processed in the same clock cycle.

The role of interrupts in a computer system

Interrupts are used when a part of the computer system needs to get the attention of the CPU or operating system. A hardware interrupt could be a key being pressed or the click of a mouse. A software interrupt could be a process such as a wizard needing the user to make a decision.

After an interrupt occurs, it is handled while whatever operation is currently running is suspended. When the interrupt has been serviced, the suspended operation is resumed. Handling an interrupt involves copying the content of the registers out of the CPU. When the suspended operation resumes, the register contents are copied back into the CPU registers.

Key terms

Bus – the motherboard has parallel metal tracks to connect different parts, allowing each bit in a byte to travel along a track side by side when transferring data between components such as the CPU and RAM.

Variable – used by program code to hold a value as the program runs.

High-level program – this is code written within most programming environments. One command in a high-level language, for example pctResult.Image = Image.FromFile("../Pics/Correct.jpg"); would compile into many lines of binary machine code. High-level programming is much easier than low-level programming as much less needs to be typed and because the keywords are much easier to understand and use.

Assessment practice 2.2

A01 A02 A05

You have a part-time job in a small computer store as an assistant customer advisor. The owner has asked for some materials that the shop can use to help explain to customers how processors work.

Produce a booklet which describes how the fetch–execute cycle works in a CPU.

Include the special registers involved in these operations along with their roles in the fetch–execute cycle.

Plan

- How will I approach the task?
- Are there any areas I think I may struggle with?
- What resources do I need to complete the task? How can I get access to them?

Do

- I can question my own learning approach.
- Am I utilising all of the support available to me?
- What am I struggling with? Do I know how to overcome this?

Review

- I can draw links between this learning and prior learning.
- I can explain what skills I employed and which new ones I have developed.
- I can make informed choices based on reflection.

C How data is represented by computer systems

You need to understand the characteristics, concepts and implications of the methods of computer data representation that are used in modern computers, and these are covered in this section.

Number systems

All of the number systems that we use have a base number which is always shown as 10 in that number system. The position of each digit is always in columns, with each column determining the scale of any digit in that column.

The lowest column for a whole number in any number system will be the base number to the power of zero. As any number to the power of zero is one, this column is always the '1s' column and no scaling is needed for any digit in this column.

The other columns have different scaling, according to the number system, with the base number at a different power for each column. We are used to using base 10, our main numbering system, which has column scaling (from low to high) of 10^0 (1s column), 10^1 (10s column), 10^2 (100s column), 10^3 (1,000s column) and so on. The base 10 numbering system is called denary.

The number 289 in denary is made from 9 in the 1s column, 8 in the 10s column and 2 in the 100s column so this number is actually 9 + 80 + 200.

The use and interpretation of number systems used in computer systems

In this section, you will look at how number systems are used and interpreted in computer systems.

Units of digital data

All parts of a computer system use one of two values to hold or communicate data or code. The two values vary according to where they are, for example, RAM has electricity or no electricity, an HDD has tiny spots magnetised to north or south, an optical drive has tiny spots that reflect or absorb a laser beam, copper network cabling has tiny bursts of electricity or no electricity.

These two values are thought of as 1 or 0, sometimes true or false. This unit of digital data is called a bit (from **b**Inary dig**it**). A bit offers a choice of only two values, but, for most purposes, a wider choice of values is needed. The basic unit of digital data is the **V**, which is made from 8 bits, giving 256 different possible combinations.

Other units of digital data use multiples of approximately thousands. These are kilobyte (1,000 bytes), megabyte (1,000,000), gigabyte (1,000,000,000), terabyte (1,000,000,000,000) and petabyte (1,000,000,000,000,000). We round these off to multiples of 1,000s whereas the binary equivalents that are actually used by systems are 1024s.

Tip

The commonly used abbreviations for storing or using multiples of bytes are: kilobyte (KB), megabyte (MB), gigabyte (GB), terabyte (TB) and petabyte (PB).

Communications are usually measured as multiples of bits and are shown using lower case letters as follows: kilobit (kb), megabit (mb) and gigabit(gb).

Binary and hexadecimal

Everything inside a computer is logically a 1 or a 0 which makes it difficult for us to understand or communicate data, such as a byte made from 01000001.

Binary is the numbering system with base 2, with column scaling (from low to high) of 2^0 (1s column), 2^1 (2s column), 2^2 (4s column), 2^3 (8s column) and so on. There are only two possible digits in binary, which makes it a good match to the data actually inside a computer.

The earlier example of a byte made from 01000001 can be expressed as 65 in denary as there is just 1 in the 1 and 64 columns and 0 in the rest (1+64=65). Obviously, 65 is much easier to write, read and understand than 01000001 as well as taking up a lot less space on a screen or paper.

IT professionals who work with binary use the hexadecimal system, rather than the denary, as it offers a much better representation of binary. Hexadecimal is the numbering system with base 16, with column scaling (from low to high) of 16^0 (1s column), 16^1 (16s column), 16^2 (256s column), 16^3 (4096s column) and so on.

Hexadecimal looks complicated, but, in practice, only the 1s column is actually used for binary conversions. This column can contain any number between 0 and 15 (in denary). Only one digit can be in a column, so letters are used for values of 10 (A), 11 (B) and so on to 15 (F).

Four binary digits can hold a range of values from 0 (0000) to 15 (1111) which is an exact match for a single hexadecimal digit which can represent these as 0 (0000) to F (1111).

Any byte can be represented as a two-digit hexadecimal number with a range of possible values from 0 (denary), 00000000 (binary), 00 (hexadecimal) to 255 (denary), 11111111 (binary), FF (hexadecimal).

PAUSE POINT Binary is a numbering system with only two valid digits, 0 and 1. Any collection of binary digits can be converted into a denary or hexadecimal number by including column headings and then adding together every column heading with a 1 under it. What is a base number? Why do column headings work for numbering systems?

Hint Research the BBC Bitesize website on binary addition to check that you are able to add two binary numbers. Translate the numbers into denary (base 10) to check your answers.

Extend What would be the column headings for a base 8 numbering system? Convert two numbers into base 8 and then add them together. Translate your answer back to base 10 to check that it is correct.

To convert a byte to hexadecimal, divide it into two four-bit numbers (nibbles) and then convert each into a hexadecimal digit. The byte of 01000001 in the earlier example can be expressed as 41 in hexadecimal.

Discussion

A two-digit hexadecimal number can represent any byte consisting of 8 bits. With a peer or in a small group, discuss the maths behind converting the upper nibble into a hexadecimal digit in the 16s column.

Binary coded decimal (BCD)

Another method for using bytes to hold denary numbers is BCD. In this system, each nibble (half of the byte) is treated as a denary digit. Using BCD, 01000001 would be 41. 11111111 would be invalid, as 99 (10011001) is the largest valid number possible in a BCD byte.

The use of binary arithmetic (including BCD) to perform calculations

Binary is a numbering system, so can be used to perform calculations. Remember: you can always check your answers by converting the binary to denary to confirm that the result is the same.

Addition

Binary addition between two numbers is very easy and just like using denary. Start from the 1s column at the right of the number and add the pair of digits, carrying to the next column if necessary.

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
						0	1	
					1			**Carry**

1 + 1 = 10
Zero goes in this column and 1 is carried over to the next column

1 + 0 = 1

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
					1	0	1	
				1	1			**Carry**

Here we need to add 1 + 1 plus the 1 carried over
1 + 1 + 1 = 11
So 1 goes in this column and 1 is carried

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
				0	1	0	1	
			1	1	1			**Carry**

1 has been carried over so in this column
1 + 0 + 1 = 10
So 0 goes in this column and 1 is carried

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	1	1	**15**
0	0	1	0	0	1	1	0	**38**
0	0	1	1	0	1	0	1	**53**
			1	1	1			**Carry**

0 + 1 = 1

0 + 0 = 0 plus the carried over 1 makes this 1

Result is 53
32 + 16 + 4 + 1 = 53

Figure 2.4: Adding two 8 bit numbers: 00001111 and 00100110 – all addition shown is binary

Subtraction

Binary subtraction can be carried out using two's complement. The number to be subtracted (subtrahend) is converted into two's complement then added to the number it is to be taken away from (minuend). An example using denary would be 35 – 12 where the subtrahend (12)

is converted into the negative (–12) then added to the minuend (35): 35 – 12 gives the same result as 35+ –12 with 23 the answer for both methods.

Key term

Two's complement – this method is used in binary arithmetic to represent negative numbers. To convert a binary number into two's complement, firstly invert the number (swap 1s for 0s and 0s to 1s) then add 1 to this inverted number to calculate the two's complement. The same operation (invert then +1) is used to convert a two's complement number back to binary.

Two's complement is similar to a negative number, but, when the calculation is made, a 1 should overflow from the high end (left) of the answer. This is expected and the 1 is ignored. If there is no overflow, the answer is negative, shown as two's complement, so needs to be converted back to binary.

128	64	32	16	8	4	2	1	
0	0	1	0	0	0	1	1	**35**
0	0	0	0	1	1	0	0	**12**

First step is to convert the subtrahend to twos complement (invert the bits and add 1)

128	64	32	16	8	4	2	1	
0	0	0	0	1	1	0	0	**12**
1	1	1	1	0	0	1	1	**invert**
0	0	0	0	0	0	0	1	**Add 1**
1	1	1	1	0	1	0	0	**2s complement**
				1	1	1		**Carry**

Now add the 2s complement number to the minuend

128	64	32	16	8	4	2	1	
0	0	1	0	0	0	1	1	**35**
1	1	1	1	0	1	0	0	**2s complement**
0	0	0	1	0	1	1	1	**23**
1	1							**Carry**

There is a carry bit on the last digit which is ignored

The result is 00010111 which in denary is 23

Figure 2.5a: Subtraction. In this example 12 (the subtrahend) is subtracted from 35 (minuend)

Multiplication

Binary multiplication is the same as with denary, but often easier to carry out. If denary is used to multiply 370 (multiplicand) by 12 (multiplier) you could multiply 370 by 10 then add this part-result to 370 multiplied by 2: 3700 + 740 = 4440.

To carry this out in binary, you have 0101110010 (multiplicand) by 1100 (multiplier). Each 0 in the multiplier can be disregarded as, when the multiplicand is multiplied by 0, the part-result will be 0. Each 1 in the multiplier needs to produce a part-result, all of which will be added together for the answer.

```
                0 1 0 1 1 1 0 0 1 0
X                           1 1 0 0
-----------------------------------
                0 0 0 0 0 0 0 0 0 0
              0 0 0 0 0 0 0 0 0 0
            0 1 0 1 1 1 0 0 1 0
          0 1 0 1 1 1 0 0 1 0
-----------------------------------
          1 0 0 0 1 0 1 0 1 1 0 0 0
```

Where the multiplier digit is zero, all the results in the part result are zero

All the part results are added up to give the answer

Result is 1000101011000 Which is 4,440 in denary

Figure 2.5b: Multiplication

Division

For the calculation 162 ÷ 6 in denary using the ‘bus stop’ method, we write the dividend (162) under the bus stop and the divisor (6) before it. Starting at the left, the divisor cannot be divided into the first dividend digit (1), so we write a 0 above the bus stop for that column then move right. The divisor can be divided into 16 twice, so we write a 2 above the bus stop for that column and then carry out the division, which leaves a remainder of 4, so we move to the right to bring down the next digit from the dividend, giving 42. The divisor can be divided into 42 seven times, so we write a 7 above the bus stop for that column and then, as we have reached the lowest digit of the dividend, the calculation is complete (see Fig 2.6).

	0	2	7
6	1	6	2
	-1	2	
		4	2
		-4	2
			0

Figure 2.6: Denary division using the bus stop method

Binary division can be carried out using the bus stop method.

For the calculation 162 ÷ 6 we need both the dividend (162) and the divisor (6) in binary. Starting at the left, if the divisor cannot be divided into the dividend, write a 0 above the bus stop for that column then move right. If the divisor can be divided into the dividend, write a 1 above the bus stop for that column and then carry out the division and move right (see Fig 2.7).

	128	64	32	16	8	4	2	1	
	0	0	0	1	1	0	1	1	
1 1 0	1	0	1	0	0	0	1	0	162
Divisor (6)		-1	1	0					
		1	0	0	0				
			-1	1	0				
			0	1	0	0	1		
					-1	1	0		
					0	1	1	0	
						-1	1	0	
						0	0	0	

Figure 2.7: Binary division using the bus stop method

The use of binary to represent negative and floating point numbers

There are many methods used inside computer systems to represent numbers using the binary system, including negative and floating point numbers.

Negative binary numbers

Two’s complement has been used to represent binary negative numbers since the earliest microprocessors, because it enabled subtraction to reuse other circuits already in the CPU for adding numbers and inversion using NOT gates.

Link

For more about NOT gates, see ‘Boolean logic’.

Floating point binary numbers

The use of floating point binary numbers to represent very large and very small numbers is similar to the standard form (also known as scientific notation) that is used with denary numbers where the number is shortened to a **mantissa** with a single digit followed by a set number of decimal places and an **exponent** to remember how many places the decimal point moved.

Using standard form, 27,563,498,208 would be written as 2.76×10^{10} (this can also be written as 2.76E10) with 2.76 representing the number to two decimal places and $\times 10^{10}$ showing that the decimal point moved 10 times to between 2 and 7 in 2.76. Accuracy is lost with standard form because to convert 2.66×10^{10} back we would get 27,600,000,000 as there is no way of recovering the detail lost in rounding the mantissa to two decimal places.

A floating point binary number has the mantissa normalised to 0.1 (1.0 for two’s complement negative numbers) with a set number of digits, and the exponent shows how many places the binary point has moved to normalise the mantissa.

Key terms

Mantissa and exponent – these are used to represent very large or very small numbers in a small space using floating point or scientific notation. The mantissa is the number part and the exponent shows how many places the point has moved, e.g. 23,467,334 in denary can be shown as 2.35×10^7, where 2.35 is the mantissa and 7 is the exponent.

Text representation

Writing text is a very common use for computer systems. Text representation standards are needed so that text and documents can be shared and viewed on the many different computer systems in use today.

The purpose and implications of using codes to represent character sets

Bytes are used in every computer system. These are made from 8 bits which can be arranged in 256 different combinations. The world of computing uses a common set of codes for the alphabet and numbers. Together these take up 62 (0–9, A–Z, a–z) of the 256 possible codes in a byte.

The 8 bit code for a character can be represented in denary or hexadecimal to give a sequence. The letter A is held in memory as 01000001 and can be represented as 65 in denary or 41 in hexadecimal. The letter B is 01000010 (66 or 42), the letter C is 01000011 (67 or 43) and so on.

The difference between upper and lower case characters is bit 6, set to 1 for lower case or 0 for upper case so the code for a is 01100001, 97 or 61, for b is 01100010, 98 or 62, for c is 01100011, 99 or 63 and so on.

This structure makes it easy for computer systems needing a user ID or word processors to recognise upper or lower case versions of a character by ignoring bit 6. Conversion between upper case and lower case is also easy by setting bit 6 to 0 or 1.

Discussion

Masking is used to convert an alphabetic character between upper and lower case using AND and OR operations between the character code and the mask. With a peer or in a small group, discuss the mask and logic operations needed to convert a character to upper case or to lower case.

Using codes to represent character sets offers a lot of possibilities for the font used for the character set. The appearance of numbers and alphabetic characters is easily changed using a different font. A font can also make use of the other codes for shapes or accented characters.

The features and uses of common character sets

In this section, you will explore the features and uses of two common character sets, ASCII and UNICODE.

ASCII

The ASCII (American Standard Code for Information Interchange) character set is used for the English language and is based on up to 7 bits with 128 different characters, which include control characters such as CR (carriage return), ESC (escape) and so on. Extended ASCII uses 8 bits.

UNICODE

ASCII has been in widespread use since the 1960s, but UNICODE has been the character set used by Windows® and most websites since the mid-1990s. The UNICODE character set uses between 1 and 4 bytes for each character, giving over a million possible characters, which makes this a good choice for systems using several languages.

UNICODE is compatible with ASCII because they share the same codes for numbers and the alphabet.

Image representation

Images are used in many ways by computer systems. The types of image used by computer systems include photographs, diagrams and animations. Bitmap/raster images are best used for photographs and vector images are best for diagrams.

How bitmap/raster image data is stored and represented within a computer system

Bitmap and raster are both terms used to describe images made from millions of dots (pixels), each of which has a colour. (There are no practical differences between the terms 'bitmap' and 'raster'). Each pixel is represented by a binary number. In a pure black and white image just a single binary digit would be sufficient to represent the two possible states of the pixel (black or white). For colour images large binary numbers are used to represent a range of colours. Using 8 bits to represent each pixel provides 256 possible colours. The number of bits used is called the colour depth, which is covered in the next section.

Images taken by a mobile phone, digital camera or through a scanner are raster images. Raster images are best used for photographs as their images are in the same form as captured by these devices.

Vector images are very different from rasters because they are made from coordinates that define the placing of objects that have properties such as line thickness and fill patterns, which are based on mathematical formulae. These images are better suited for drawings and diagrams than photographs.

Raster images can be converted into vector images although this is a time-consuming process and may even be impractical for some photographs due to the amount of fine detail.

The impact of image resolution on the way images are stored and represented

Image resolution is shown as two numbers, for example 1920×1080 which means 1080 rows of 1920 pixels. The bigger the number the more detailed is the image, which has more pixels and a larger file size when saved to backing storage.

Vector images typically have much smaller file sizes due to the limited amount of stored information for this type of image compared with the need to store the colour of every pixel in a raster image.

Printers and displays use different technologies to make images, but they all create dots on paper or other media. The most efficient use of storage space is made when the resolution of the image matches that of the output device. In a raster image, fine detail that cannot be displayed or printed wastes storage space.

Designers of web pages using images should consider the purpose and audience of the web page when deciding how high the resolution needs to be for the images they are using. There is no point in using a high resolution image which will then be scaled down by a web browser so that it fits onto the web page because it would slow down the loading of the page.

The quality of the image is not just a matter of the resolution, but also of how small and closely packed the pixels are. This is measured in dots per inch (dpi), with a higher number of dpi meaning that the quality of the image is higher.

The impact of sample/bit depth on the way that image data is stored and images are displayed

The bit depth means how many colours are sampled when the image is input. 256 colours use 1 byte for each pixel, which is also known as 8 bit colour depth. Other popular colour depths are 16 bit (64K colours), 24 bit (16.7 million colours) and 32 bit (16.7 million colours with one of the four bytes used for transparency).

There is a direct relationship between the colour depth, resolution and storage needs, because each of these aspects of the image affect the number of bytes used.

Vector image formulae are recalculated when the image is displayed or resized so there is no loss of quality at a different scaling. Changing the size of a raster will affect image quality. If the image is to be reduced, the software needs to remove pixels. If it is to be enlarged the software will have to 'make up' extra detail or show pixels as blocks of colour.

The effects of compression on image data

The size of storage space needed by a bitmap can be reduced through **compression**, which can be lossy or lossless.

- Lossy: Lossy compression loses some of the image detail when it is compressed, as with JPG image files. Data that is not needed, such as small differences between colours and any additional metadata, is removed. Therefore editing a JPG, saving and then re-editing it often reduces the image quality.
- Lossless: Lossless compression reduces the image size without any loss of quality, although the amount that the file size is reduced by is less than with lossy compression. For example, PNG image files use techniques to analyse and reduce the image by blocking pixels of the same colour together to store their colour, number of pixels and location in the image rather than having the colour repeated for each pixel.

Key term

Compression – is a technique used to make files smaller. Compression can be a feature of a file format or can be implemented by utility software such as WinZip.

Link

See the section on 'Types of compression'.

Assessment practice 2.3

A01 A02

You have been asked to prepare a presentation that can be used as a rolling display for a trade stand at a business fair to help promote a new range of storage devices.

The presentation is to be divided into three sections, each with an introduction slide followed by slides detailing the technical content.

Section 1: Binary representation of data

Explain how binary bits can be combined together into bytes, megabytes, gigabytes and terabytes with examples of where each could be found inside a computer system.

Section 2: Text representation of data

Explain how ASCII and UNICODE are used.

Section 3: Image representation

Explain how the quality of a digital image can have an impact upon the storage size.

Plan

- What am I being asked to do?
- Do I need clarification of anything?
- Do I have any existing knowledge for the task at hand?

Do

- Am I confident that I know what I am doing and that I know what it is I should be achieving?
- I can set milestones and evaluate my progress and success at these intervals.
- I can make connections between what I am reading/researching and the task, and identify the important information.

Review

- I can explain how I approached the task.
- I can explain which elements I found easiest.
- I can explain which elements I found hardest.

D How data is organised on computer systems

You need to recognise how data is used in computer systems. In this section, you will explore the characteristics and implications of methods of organising data in computer systems, and the impact this has on computer processes.

Data structures

There are many different ways in which computer systems can structure data into different sequences for processing.

The features, applications and implications of data types used within computer systems

In this section, you will look at the features, applications and implications of different data types.

Stack

A stack is a FILO (first in, last out) data structure where items are pushed onto the stack then popped off in the reverse order. This structure is used when a subroutine is called in code to remember where the code needs to return to when the subroutine completes.

Conceptually, a stack is like a pile of plates where plates are added or removed from the top of the pile. See a visual representation of a stack in Figure 2.8.

Discussion

With a peer discuss how a stack can be used when a spreadsheet works out a complex calculation.

Queue

A queue is a FIFO (first in, first out) data structure where items are added to the queue then used in the same order. This structure is used for batch processing where items are dealt with in order, for example when printing.

Conceptually, a queue is like a conveyor belt on a supermarket checkout. See a visual representation of a queue in Figure 2.8.

Discussion

With a peer discuss how queues can be used in a modern processor.

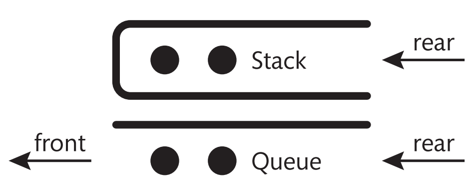


Figure 2.8: Stack vs queue

Array

An array is a type of variable used in programming where more than one item is to be held in the variable. The variable name is usually followed by subscript numbers inside brackets to define which item in the array is to be used. If you wanted to store the temperature on each day of the week you would define an array with 7 elements. Assuming the array was called Temperature, then Temperature(0) would hold Sunday's temperature, Temperature(1) would hold Monday's and Temperature(6) would hold Saturday's. There are many techniques that programmers use when coding with arrays, in particular, using iteration to loop through items in the array.

Arrays are stored in adjacent memory locations, so space is cleared and reserved for the whole array when the array is first declared in the program code.

Discussion

With a peer, discuss how an array can be used by a program to sort data.

List

A list is a data structure where an item links to another item. There are many types of list used in different ways, for example a database uses indices to define different sequences through a dataset.

Lists often use pointers in data items to point to the address of the next item in the list. An example of this technique is a binary tree which is used by database systems to store items of data as they are entered with links so that the items can be searched and brought back in an ordered sequence. For example, Fig 2.9 shows a binary tree where items were entered in this order: Jones, Mellor, Ali, Smith, McGill, Patel, Singh. Each item starts with both left and right pointers set to null (-1).

- Jones goes into memory location 1.
- Mellor goes into memory location 2.
- Mellor is compared with Jones; alphabetically more, so right null in Jones is set to 2 to point to Mellor.
- Ali goes into memory location 3.
- Ali is compared with Jones; alphabetically less, so left null in Jones is set to 3 to point to Ali.
- Smith goes into memory location 4.
- Smith is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- Smith is compared with Mellor; alphabetically more, so right null in Mellor replaced by 4 to point to Smith.
- McGill goes into memory location 5.

- McGill is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- McGill is compared with Mellor; alphabetically less, so left null in Mellor set to 5 to point to McGill.
- Patel goes into memory location 6.
- Patel is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- Patel is compared with Mellor; alphabetically more, right not null in Mellor, so pointer is followed to Smith.
- Patel is compared with Smith; alphabetically less, so left null in Smith set to 6 to point to Patel.
- Singh goes into memory location 7.
- Singh is compared with Jones; alphabetically more, right not null in Jones, so pointer is followed to Mellor.
- Singh is compared with Mellor; alphabetically more, right not null in Mellor, so pointer is followed to Smith.
- Singh is compared with Smith; alphabetically less, left not null in Smith, so pointer is followed to Patel.
- Singh is compared with Patel; alphabetically more, so right null in Patel set to 7 to point to Singh.

Memory location	Left pointer	Items	Right pointer
1	3	Jones	2
2	5	Mellor	4
3	–1	Ali	–1
4	6	Smith	–1
5	–1	McGill	–1
6	–1	Patel	7
7	–1	Singh	–1

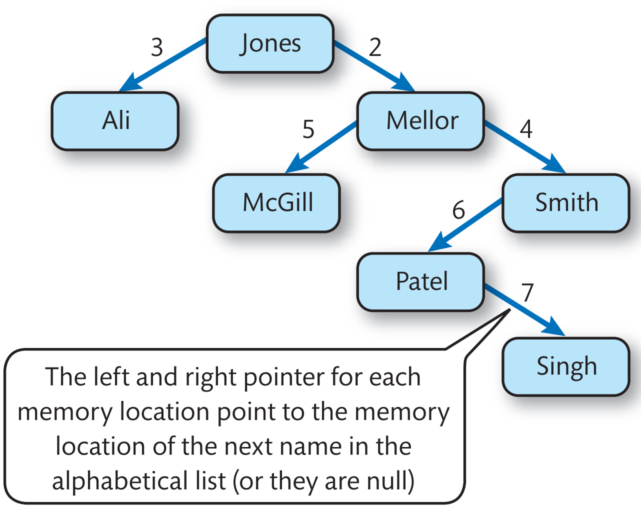


Figure 2.9: Binary tree

Conceptually, a list is a link to the next item in the list.

Discussion

With a peer discuss how a list can be used as a binary tree to structure storing and sorting data.

The use and application of data types in computer software

All computer software is created using some form of programming environment and language, such as C#. Almost all software uses variables to hold information or data when the code runs.

All programming languages allow the coder to define their variables so that each variable has a data type such as integer or string. Some programming languages also allow variables to be used with a default data type called a variant which is able to adapt to and hold any kind of data type. Variant data types carry a lot of performance-reducing overheads to cater for all the possible uses of a variable, most of which will not be needed for the variable.

Although variant data types are quicker to apply and cause fewer problems, it is much better practice to define a variable to hold the most appropriate data type for the following reasons.

- More accurate processing: Any places where the program tries to allow a wrong data type into a variable will be identified during testing, which helps to avoid errors during design or coding.
- Better use of memory: There will be no wasted space and memory will be optimised for that data type.
- Faster processing: There be no need for extra processing to examine data inside a variable before carrying out arithmetic or logical operations, which there would with a variant data type.

Data types

Programmers writing code use variables such as a name or quantity to hold data when the program is running. Best practice is to declare variables with a data type to maximise memory use and to check the expected type of data is input into the program. Variable types include:

- number when calculations are needed
- integers for whole numbers
- real for numbers with decimal places
- currency for money
- string for any mix of text and numbers
- Boolean for true or false.

The use and implications of data types in computer hardware

The different data types need to be stored and processed by computer hardware.

Storage requirements are directly linked to the data types. At the start of section C, you saw the different ways in which numbers can be stored in the memory. This also has an implication for the need for the ALU hardware circuits in the CPU to be able to correctly calculate with every possible type of data.

Indices and matrices

A matrix is a rectangular collection of numbers which can be used in mathematical operations. Matrices are used in many areas of computing including in the calculations needed for 'wire-framing', which forms the basis of virtual gaming landscapes. A 2D matrix has two rows which can be used to hold the (*x*, *y*) coordinates of a shape. These can be multiplied by other matrices to produce new (*x*, *y*) coordinates for operations such as rotation, mirroring, resizing, and so on. Indices are used to identify items in a matrix.

Matrix representation in computer systems

In this section, you will explore how matrices are represented in computer systems (including their relationship with arrays), mathematical operations using matrices and how to order a matrix.

Matrices and arrays

There are close relationships between matrices and arrays. Many programming languages store an array used by a programmer in their code as a matrix in the memory. Similarly, a programmer wishing to implement matrix techniques in their code would declare and use arrays to hold the data for such calculations.

An array can be single-dimensional (1D), two-dimensional (2D) or multi-dimensional, for example 3D. See Figure 2.10 for a visual representation of 1D, 2D and 3D arrays. There will be an equivalent number of subscripts to the dimension, so a 1D array has one subscript, 2D has two subscripts, 3D has three subscripts and so on.

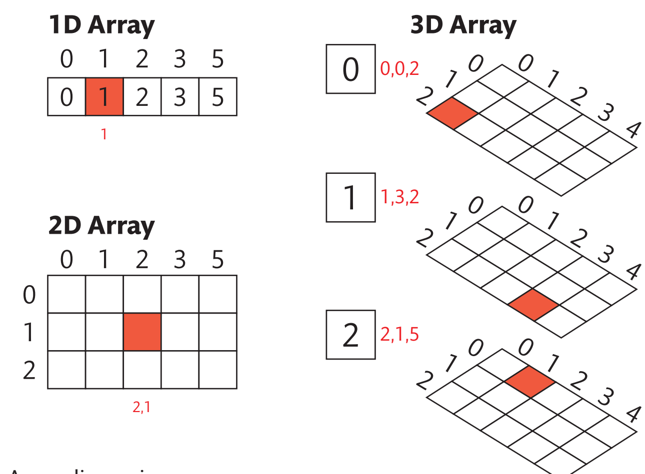


Figure 2.10: Array dimensions

Key term

Inversion – means flipping a value. In a Boolean inversion 0s are inverted into 1s and 1s are inverted into 0s. In a mathematical inversion a fraction is turned upside down so 2 would invert to $\frac{1}{2}$ and $\frac{1}{4}$ would invert to 4. This is also called finding the reciprocal of a number.

Mathematical operations

Mathematical operations include adding or subtracting matrices of the same size, multiplying matrices of different (or same) sizes and **inversion**. These mathematical operations can be used for manipulating shapes and solving simultaneous equations.

- Manipulating shapes is very easy using matrices. A matrix of (*x*, *y*) coordinates is multiplied by a 2×2 matrix with each pair of (*x*, *y*) coordinates producing a pair of transformed coordinates (see the worked example).

Worked Example

A shape is reflected in the *y*-axis using matrix multiplication. This example is shown in Fig 2.11 Reflection in the *y*-axis using matrix multiplication.

Step 1: Draw a shape on graph paper using straight lines with 5 corners. Write the (*x*, *y*) coordinates onto paper as a table with two rows

x	1	2	3	4	2
y	2	5	3	7	7

Step 2: Use matrix calculations to multiply your (*x*, *y*) coordinates by this 2×2 matrix

(a)	–1	0	(b)
(c)	0	1	(d)

Step 2a: Calculate the upper number in the answer. The formula for this is $ax + by$ with a, b shown above and (*x*, *y*) the first pair of coordinates in your table

$-1 \times 1 + \; 0 \times 2 \quad = \quad -1 + 0 \quad = -1$

Step 2b: Calculate the lower number in the answer. The formula for this is $cx + dy$

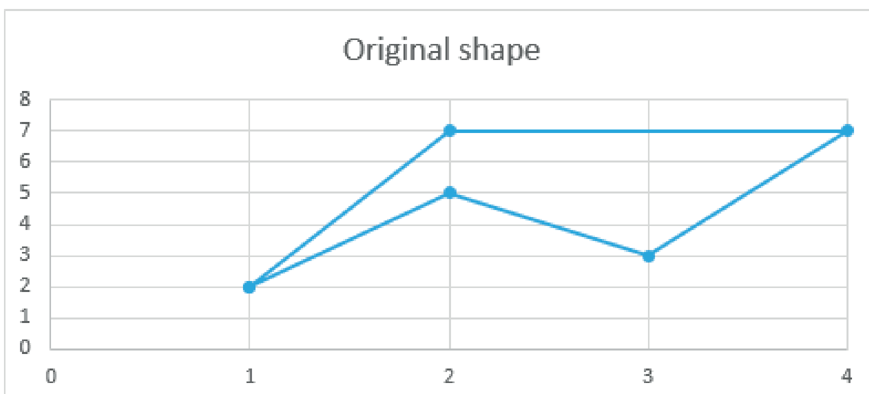
$0 \times 1 \; + \; 1 \times 2 \quad = \quad 0 + 2 \quad = 2$

Step 3: After repeating steps 2a, 2b for the rest of your (*x*, *y*) pairs you have the new (*x*, *y*) coordinates for your shape mirrored in the *y*-axis.

x	–1	–2	–3	–4	–2	–1
y	2	5	3	7	7	2

X	1	2	3	4	2	1
Y	2	5	3	7	7	2

Original XY coordinates



-1	0
0	1

2x2 matrix

Rotates 180 degrees

X	-1	-2	-3	-4	-2	-1
Y	2	5	3	7	7	2

Transformed XY coordinates

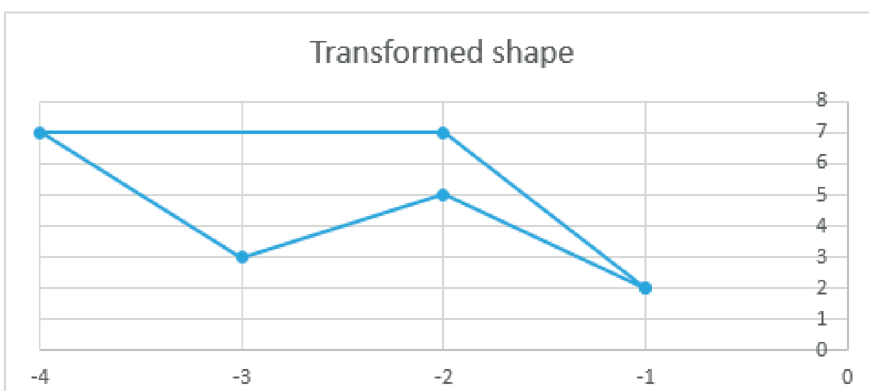


Fig 2.11: Reflection in the *y*-axis using matrix multiplication

X	1	2	3	4	2	1
Y	2	5	3	7	7	2

Original XY coordinates

-1	0
0	-1

2x2 matrix
Rotates 180 degrees

X	-1	-2	-3	-4	-2	-1
Y	-2	-5	-3	-7	-7	-2

Transformed XY coordinates

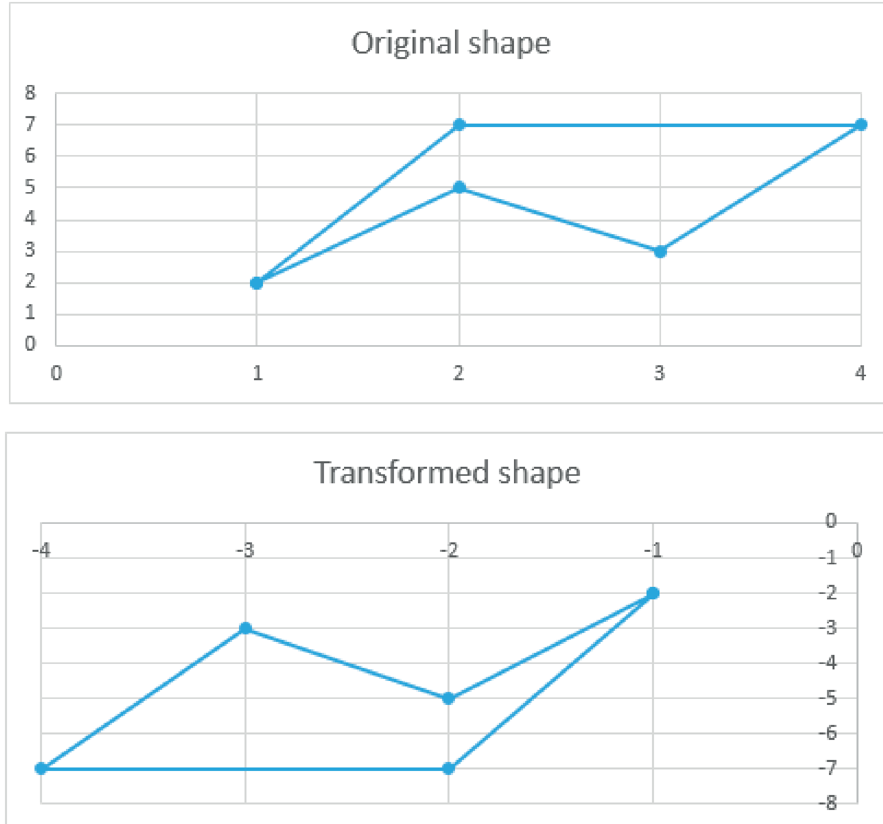


Fig 2.12: 180 degree rotation using matrix multiplication

Solving simultaneous equations is a little more complex (see the worked example).

Both of the equations terms need to be in the right sequence:

<number#1>X <+ or –> <number#2>Y = <number#3>

so, if one of the pair of simultaneous equations was $3x = 26 - 2y$ it would need to be manipulated into:

$3x + 2y = 26$

If the other of the pair of simultaneous equations was $28 = 10y - 2x$ it would need to be manipulated into:

$-2x + 10y = 28$

The simultaneous equations can now be represented as a matrix multiplication:

$$\begin{array}{c}(a)\\(c)\end{array}\begin{pmatrix}3 & 2\\-2 & 10\end{pmatrix}\quad\begin{array}{c}(b)\\(d)\end{array}\times\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}26\\28\end{pmatrix}\quad\begin{array}{c}(ax+by=26)\\(cx+dy=28)\end{array}$$

This matrix can be thought of as simple algebra:

Matrix_2×2 × Matrix_1×2 = Matrix_Known

We are trying to find out the values of Matrix_1×2; the values of the other matrices are already known, so the algebraic formula needs to be manipulated so that Matrix_1×2 is the subject:

Matrix_1×2 = Matrix_Known / Matrix_2×2

To divide Matrix_Known by Matrix_2×2 we need to calculate the inversion of Matrix_2×2 which can then be multiplied by Matrix_Known (we want to divide a number by 4, we can multiply it by the inversion of 4 which is $\frac{1}{4}$).

Theory into practice

Draw a shape with 5 corners on graph paper using straight lines. Write the (x, y) coordinates onto paper then use matrix calculations to multiply these by the 2×2 matrix shown in Fig 2.12 to calculate the new (x, y) coordinates for your shape rotated by 180 degrees.

Plot your new (x, y) coordinates onto graph paper to check your calculations.

Use Excel® to repeat this exercise. Notice how the first (x, y) coordinates are repeated in the last column of the spreadsheet so Excel® draws the last line in your chart. You can select the cells where you want the new XY coordinates then use the MMULT() array formula to calculate them. The first array in your MMULT() formula is the 2×2 matrix, the second gives your (x, y) coordinates. The MMULT() formula completes with Ctrl+Shift+Enter.

Worked Example

A mobile phone shop has two offers that interest you, the MegaSaver (£30 per month then 20p per minute) and the SuperBundle (£50 per month then 10p per minute). The MegaSaver will be cheaper for a small number of calls and the SuperBundle cheaper for a large number of minutes. At which point does the SuperBundle become cheaper than the MegaSaver?

Step 1: Convert the deals into simultaneous equations

MegaSaver	Cost = £30 + (£0.20 × Minutes)	$C = 30 + 0.2M$
SuperBundle	Cost = £50 + (£0.10 × Minutes)	$C = 50 + 0.1M$

Step 2: Convert the simultaneous equations into this sequence *A* + *B* = <number>

MegaSaver	Cost - (£0.20 × Minutes) = £30	$C - 0.2M = 30$
SuperBundle	Cost - (£0.10 × Minutes) = £50	$C - 0.1M = 50$

The calculations will work as shown above, but are easier with the decimals removed by multiplying the equations by 10:

MegaSaver	$10C - 2M = 300$
SuperBundle	$10C - M = 500$

Step 3: Convert the simultaneous equations into a matrix multiplication

$$\begin{matrix} 10 \times -2 \\ 10 \quad -1 \end{matrix} = \begin{matrix} 300 \\ 500 \end{matrix}$$

Step 4: Invert the 2×2 matrix

Step 4a: Calculate 1/(*ad* − *bc*)

$$\frac{1}{ad - bc} \qquad \frac{1}{-10 - (-20)} \qquad \frac{1}{-10 + 20} \qquad \frac{1}{10}$$

Step 4b: Swap *a* and *d*, change the signs of *b* and *c*

$$\begin{matrix} -1 & 2 \\ -10 & 10 \end{matrix}$$

Step 4c: Multiply each term by 1/(*ad* − *bc*) which is 0.1

$$\begin{matrix} -0.1 & 0.2 \\ -1 & 1 \end{matrix}$$

Step 5: Multiply the inverted 2×2 matrix by the 1×2 matrix

$$\begin{matrix} -0.1 & 0.2 \\ -1 & 1 \end{matrix} * \begin{matrix} 300 \\ 500 \end{matrix}$$

Step 5a: Calculate the upper number in the answer. The formula for this is *ax* + *by*

$-0.1 * 300 + 0.2 * 500 = -30 + 100 = 70$

Step 5b: Calculate the lower number in the answer. The formula for this is *cx* + *dy*

$-1 * 300 + 1 * 500 = -300 + 500 = 200$

Step 6: The calculated answers show that at the point where the SuperBundle becomes cheaper than the MegaSaver the cost (*C*) will be £70 and the number of minutes (*M*) is 200.

Discussion

With a peer or in a small group, discuss what transformations can be performed by multiplying different 2×2 matrices by a shape defined as a matrix of (x, y) coordinates.

Ordering a matrix

Memory locations are in a simple sequence and any memory usage to hold a matrix is applied to that sequence. The way in which a matrix can be held in memory will be either in row-major or column-major order.

With **row-major order,** the top row is held in the memory in the same sequence as the top row of the matrix, then the next row follows in the memory, and then any other rows follow afterwards. This is shown in the following example:

13	31	1	63	60
76	47	4	49	8

13	31	1	63	60	76	47	4	49	8

With **column-major order**, the first column is held in the memory, then the next column follows in the memory, and then any other columns follow afterwards. This is shown in the following example:

94	74	63	91
89	76	46	51
21	65	46	61

94	89	21	74	76	65	63	46	46	91	51	61

Assessment practice 2.4

A02 A03 A04

You are a member of a team that has been selected to produce two training sessions for students in their first year of programming at local centres.

The first session is to describe and explain four of the data structures commonly found in computer systems with an example of how each of the following could be used:

- stack
- queue
- array

The second session is to describe and explain how matrices can be used in computer systems to solve problems and to manipulate graphics.

Produce a set of handouts which:

- demonstrate how to solve simultaneous equations using matrices for a real-world problem
- use a spreadsheet model to demonstrate how shapes can be manipulated.

Plan

- What resources do I need to complete the task? How can I get access to them?
- How much time do I have to complete the task? How am I going to successfully plan my time and keep track of my progress?
- What aspects of the task do I think will take the most/least time? How will I balance these?

Do

- What strategies am I employing? Are these right for the task? Are they working? If not, what do I need to do to change this?
- I understand my thought process and why I have decided to approach the task in a particular way. I can explain this reasoning when asked.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.

Review

- I can draw links between this learning and prior learning.
- I can explain what skills I have employed and which new ones I have developed.
- I can make informed choices based on reflection.

E How data is transmitted by computer systems

Communication is everywhere in the modern world, with data being transmitted between millions of computer systems 24/7. You need to understand the concepts, processes and implications of data transmission in and between computer systems, and these will be covered in this section.

Transmitting data

There are many methods used to transmit data in and between computer systems. In this section, you will explore the types of communication channel, methods of connecting devices and transmitting data and different types of transmission, along with the protocols used to control data transmission.

Types of communication channel

There are a number of different communication channels to choose from depending on the situation.

- **Simplex** can only be in one direction. Examples include a sensor sending temperature data into a control system or a television transmission.
- **Half-duplex** must be used in both directions, but only in one direction at a time. Examples include data in an old bus network topology or using walkie-talkie radios.
- **Full-duplex** uses both directions at the same time. Examples of this include current network cabling or voice telephone calls.
- **Point-to-point** is when there is direct communication between two computer systems.
- **Multi-drop** is when there is communication from one computer to several other computer systems.

Discussion

With a peer or in a small group, discuss how an organisation can make use of each type of communication channel.

Methods of connecting devices and transmitting data across and between computer systems

Devices can be connected using wired or wireless technologies in order to transmit data across and between computer systems.

Wired connections

Wired network connections could be either using a **USB** cable or a network. USB is usually used for short distance point-to-point transmissions, such as synchronising data between a tablet and a PC which might be standalone or attached to a network. Wired network connections are standard for workstations in organisations and are used for a wide variety of tasks including data entry.

Wireless connections

Wireless connections could be either using **Bluetooth** or WiFi technologies. Bluetooth is usually used for short distance point-to-point transmissions, such as synchronising data between a tablet and a PC which might be standalone, attached to a network or between mobile devices. WiFi connections are standard for laptops used in organisations where the user needs to log into the network. Once connected, a laptop can be used as a workstation.

Key terms

USB – a universal serial bus is a widespread socket or port found on computers, televisions, DJ decks, satellite-navigation systems and many other devices.

Bluetooth – this is a short-range wireless standard that can connect a mobile phone to a headset or transfer data such as music between devices.

The selection of connection methods to fulfil specified tasks and functions

For most organisations, the selection of connection method is a choice between WiFi or wired (cables) which could be either **copper** or **optical**.

A wireless WiFi connection is the obvious choice for tablets and laptops which need to connect to the main network. In some industries, it is common to use laptops with a wireless network connection both remotely and in the office so that it is easy to use laptops during meetings.

Cabled copper connections are a common choice for PC workstations as they are cheap, have good data throughput speed and are easy to install. Optical connections are common for **switch** to switch communications as their excellent speed justifies the extra cost but they are more difficult to install as they are harder to bend around corners.

Key terms

Copper – in this context, this is the name given to cabling using any metal to conduct electrical signals.

Optical – in this context, this is cabling that uses thin glass fibres to pass information using flashes of light. Optical cabling usually transmits data faster than copper cabling.

Switch – a box with a number of ports for network cables to plug into. The circuits inside the switch send any data they receive to the port holding the cable that connects to the device where the data is needed.

Key terms

Synchronised – computer systems each have clocks inside their circuits to provide timing pulses when moving data. A synchronised signal needs the clocks at each end of a data transmission to beat together.

Asynchronous – computer systems each have clocks inside their circuits to provide timing pulses when moving data. Asynchronous signals do not need the clocks at each end of a data transmission to beat together.

Asynchronous and synchronous data transmission

When computer systems communicate data, they send each byte of data as 1s and 0s (bits) which then get reconstructed back into bytes by the receiving system. Using a copper connection, a 1 could be a very quick electrical pulse of 5 volts with a 0 at no volts. Timing is very important, as a transmission of 111 would be a pulse of 5 volts lasting 3 times as long as a transmission of a single 1 (this is illustrated in Figure 2.13).

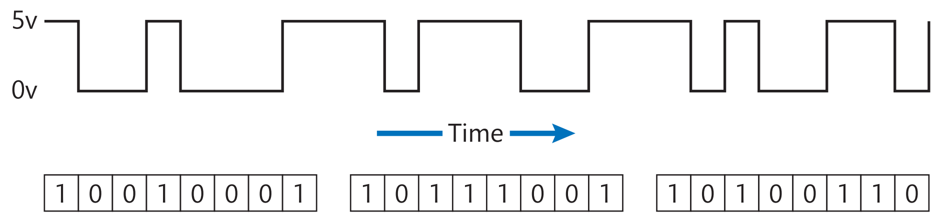


Figure 2.13: Data transmissions

With a synchronous data transmission, the device at each end of the connection has a **synchronised** clock. This means that the timings are easy to implement in components such as a computer motherboard.

Asynchronous data transmissions are very common in modern communications, as few systems need to synchronise their clocks when data is sent as packets. Each packet travels independently to the receiving system which is individually synchronised when it arrives (see Figure 2.15)

Parallel and serial transmission

Any computer data transmission will send data bytes between computer systems. A byte is made from 8 bits which can either be sent in parallel or serially. A parallel connection has a number of connections, side by side, usually a multiple of 8, so that the byte(s) can travel in a single transmission. With a serial connection, the bits in a byte travel after each other, one bit at a time.

A **parallel transmission** needs a connection for each bit. This would be a separate wire in a cable or a separate track in a circuit such as a motherboard. Parallel cables for connecting a printer to a PC were developed in the 1970s and were widely used until USB took over.

Serial transmission requires a simpler cable with just a pair of wires being needed to send a bit. In the days of parallel printer cables, a slow standard named RS232 was used. USB has been the standard since the late 1990s for serial transmissions and is now used everywhere. USB has been developed further into faster variants with USB 3.0 being the current standard.

Serial cables are a lot quicker than parallel cables. This is because there are timing issues in parallel cables because a complete byte must be received before the next can be transmitted, which is more difficult with longer cables as the bits arrive at slightly different times in the parallel wires.

PAUSE POINT Why are parallel transmissions not used everywhere? Surely sending data byte(s) by byte(s) rather than bit by bit must be faster? What are the technical reasons behind the superiority of serial communications?

Hint Research Centronic printer cables and RS232 serial cables. Compare their technical specifications.

Extend Search for 'why is parallel slower than serial' to read around the subject. In particular, find out how hard disc connections in a PC have developed through the years.

Use of packets in transmitting data

Data transmitted between systems is usually divided into packets, which are individually routed to their destinations and are rebuilt into data by the receiving system. The structure of a data packet varies according to the protocols and network standards used for the transmission, but will normally include a header, data payload and footer/trailer (see Figure 2.14).

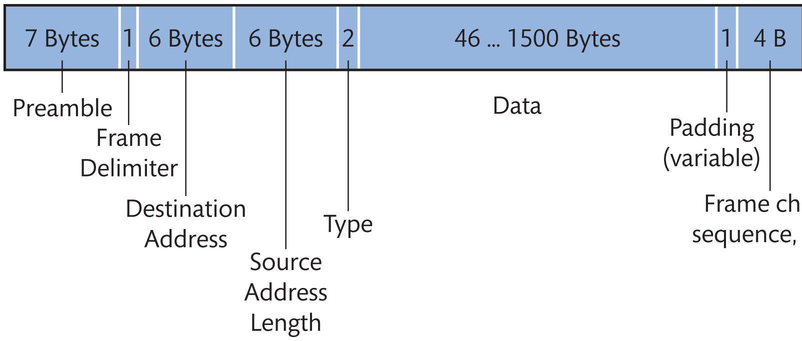


Figure 2.14: Data packet

The header section will include a preamble which states where the packet is going to (destination), where it came from (source), protocol and packet number (see Figure 2.15). The preamble is a regular 1010 so the receiving system can synchronise timing with the packet with a delimiter (11) at the end of the preamble to mark the start of the packet. The source address is needed so that the destination system can request a resend if a corruption occurs.

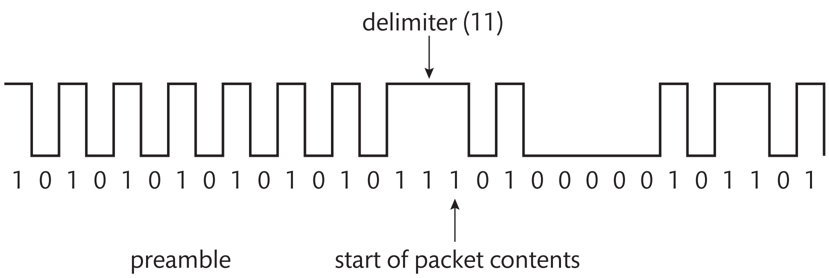


Figure 2.15: Data packet preamble

The data payload is by far the largest part of the packet. This is the data carried by the packet.

An error check is included at the end of the packet to identify any corruption that may have happened during transit. The source system uses the bytes in the packet to calculate a number which is placed at the end of the packet. The destination system carries out the same calculation on what it receives, and this is compared with the check value. If an error is detected, it triggers an automatic request for retransmission.

Packet switching

Packet switching is used in modern communication systems for the internet and telephone calls. Data is split up into packets rather than being sent in a continuous stream by the router. Not all the packets will necessarily follow the same route. At the receiving end packets are reassembled into their original order. Old telephones used circuit switching where a direct connection was opened up for each phone call, but these old mechanical exchanges have now been replaced with digital packet-switching systems. Packet switching is more efficient than circuit switching: it allows data to be sent by multiple routes and prevents large data transfers 'hogging' a circuit.

Protocols used to govern and control data transmission

Protocols are the rules agreed when computer systems transmit data. Without a protocol, the receiving computer would not be able to make any sense of the transmission.

The protocol determines the speed of transmission, error checking, the type of information, the packet structure, how data is forwarded and everything needed to successfully transfer data.

TCP/IP are commonly used protocols that allow you to surf the internet. The internet protocol (IP) enables a browser such as Internet Explorer to find a web page. The transmission control protocol (TCP) handles the movement of data between the web page and the browser.

The features, applications and implications of encryption

Encryption is used to protect sensitive data by scrambling the contents of a file so that it cannot be opened or read by unauthorised users. Encryption is a technique that is used to transmit and store sensitive information.

Simple encryption ciphers

Two simple encryption ciphers are the Caesar cipher and the Vigenère cipher.

Caesar cipher

The Caesar cipher shifts characters a number of places. For example, a start letter of J would mean that A is written as J, B as K and so on.

This method of communicating private information was used by the famous ancient Roman, Julius Caesar, to encrypt messages that he sent. Although this method of encryption is very simple, it was effective because the encrypted messages looked as though they were written in a foreign language.

Encrypting a message using a Caesar cipher can be done using a pair of wheels, where one of the wheels is moved so that the starting letter lines up with 'A' and the encrypted letters can now be read off.

Figure 2.16: The Caesar cipher works by shifting characters

A computer system using a Caesar cipher could use an encryption key, rather than a pair of wheels, to define how many times to shift each letter. For example, if the encryption key is 00000101 in binary, this would be 5 so 'B' would be encrypted to 'G', which is a shift of five characters.

Vigenère cipher

The Vigenère cipher uses a table to look up the ciphered characters.

This cipher dates back to 1553, but is named after Blaise de Vigenère who popularised it three hundred years later. Vigenère ciphers were used in many situations, such as the American Civil War.

This cipher is an extension to the Caesar cipher which uses a keyword to vary the shifting characters of the original message.

Encrypting a message using a Vigenère cipher can be done using a lookup table, as shown in Figure 2.17. If the keyword was BTEC, the first letter of the message would be looked up from the column of that letter in the B row, the second letter in the T row and so on. For example, MEETYOULATER with a key of BTEC would be NXIVZHWMTXGS.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2	B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
3	C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
4	D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
5	E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
6	F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
7	G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
8	H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
9	I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
10	J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
11	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
12	L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
13	M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
14	N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
15	O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
16	P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
17	Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
18	R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
19	S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
20	T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
21	U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
22	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
23	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
24	X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
25	Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
26	Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 2.17: The Vigenère cipher works using a lookup table

Encryption used in computer systems

Both Caesar and Vigenère ciphers are easily cracked by using language analysis techniques to find the letters which naturally occur more frequently, or even by a simple brute force method where the message is repeatedly output with characters shifted a place until a coherent message appears. For a simple Caesar cipher this would always be less than 26 repetitions.

Much more sophisticated techniques are used in computer-based encryption. There is a lot of processing available to the sending and receiving systems, so complex **algorithms** are easily implemented. These algorithms use one or more keys to calculate encrypted characters, rather than sampling by shifting along the alphabet.

Key term

Algorithm – a solution to a programming problem that is used to help plan the code that will be written to solve the problem.

Symmetric key encryption

The symmetric key encryption method uses the same key for encrypting the data as for decrypting. The algorithms using this method may transform the key before encryption to make the cipher more difficult to crack.

The major disadvantage of this method is that both the sender and receiver need the key.

Public key encryption

Public (also known as asymmetric) key encryption uses two keys: a public key and a private key. This technique is more popular than symmetric keys as the processing can happen automatically.

When an encrypted message is to be sent to a computer, the receiving system sends a public key to the sender. The sending system uses the public key to encrypt the message, but only the private key (known only to the receiver) can be used to decrypt the message.

Both the public and private keys are calculated from the same beginning in a way that cannot be reverse engineered to find one key from the other.

Public key encryption is used when you visit a secure (https) website to carry out an operation such as paying for products or services or looking at your bank account details.

Link

For more about encryption methods, see *Unit 7: IT Systems Security and Encryption*.

Types of compression

There are two types of compression, **lossy** and **lossless**. The main difference between the two types of compression is that lossy compression involves the loss of some detail in the file when it is then decompressed, whereas lossless compression involves no loss of detail so, when decompressed, you get back an exact version of the original.

Link

For more information on the two types of compression, see 'The effects of compression on image data'.

The applications and implications of data compression

Data compression is a technique that is used to reduce the size of files. This is useful as it reduces how much storage space is needed for files and also improves transmission speeds between devices because smaller files will travel through a communication link faster than larger files.

There are many methods of compressing data, but the main ways are through the choice of file type or by using a utility to compress files.

File type

The file type chosen can have a large impact on file size. Music and video files can be very large in their original form but can be reduced in size quite dramatically by saving them in an appropriate file format such as MP3 for music or MP4 for video (which are both lossy compressions). Using these smaller file sizes for music and video makes both storing and transmitting them much easier and quicker compared with a non-compressed format such as WAV for music or AVI for video.

Link

For more about image file types, see 'Image representation'.

Compression utility

Windows® comes with an included compression utility which produces a zip file from file(s) and/or folder(s). This is an effective method of compressing documents and is compatible with most other computer systems.

PAUSE POINT Symmetric key encryption is used a lot less than public key encryption. Why is this? How does public key encryption actually work?

Hint Reread the section in this unit on this subject, then carry out some internet research on the keys used by each of these encryption methods, and find out what the keys do.

Extend Public key encryption uses two keys, a public key and a private key. How are these related? Why is it not possible for another system that has been able to capture the public key to decrypt data sent out using that key?

Other compression utilities such as WinRAR are available, which has the added feature of password protection. If the compressed data is sent to another computer system, a compression utility that is compatible with the compressed file will be needed to be able to decompress and then use the data.

To compress or decompress data files requires processing power and memory, and so will take some time to complete. This is not usually an issue, but might become one on an old, low specification computer system.

Error detection

Error detection is crucial to finding any problems in the thousands of millions of 1s and 0s (the data) held in and transmitted through a computer system every day.

Methods used to detect errors in data transmission

There are a number of different methods of error detection which are used in different parts of the computer system. Data is often checked by more than one error detection method to reduce the number of errors.

Parity schemes

Parity schemes are used in the RAM to check the accuracy of each byte. A byte has 8 bits and using a parity scheme another bit is allocated to the byte to help check for errors. It does this by counting the number of bits set to 1 in the byte then using the parity bit to make the number of 1s an odd or even number.

An odd parity scheme would use the parity bit to make the number of 1s in the byte into an odd number. For example the parity bit for 01101110 would be 0 because there are 5 (an odd number) bits set to 1 already and for 01101111 would be 1 because there 6 (even number) bits set to 1 so another 1 makes it an odd number of 1s.

If 01101110 0 (odd parity) was transmitted and a corruption occurred, 01101010 0 could be received. As this has an even number of 1s (i.e. 4), the receiving device would detect the error and request a retransmission.

Similarly, an even parity scheme would use the parity bit to make the number of 1s in the byte into an even number, for example the parity bit for 01101110 would be 1 and for 01101111 would be 0. However, parity schemes are not very good at detecting multiple bit errors, since if two bits are received as 0s rather than 1s, the parity will remain correct, despite the error.

Checksum

Checksums are used at the end of a data transmission to check for errors. A calculation is carried out on the transmission before sending and the same calculation is worked out by the device that receives the data. Any difference in the checksum indicates an error so a request is sent back to the sending device requesting retransmission.

A simple checksum treats each byte in the data transmission as a number and adds them together.

Repetition schemes

Repetition schemes simply consist of transmitting the data more than once. Any difference in the repeated transmissions indicates an error.

Cyclic redundancy check (CRC)

CRC is a type of checksum placed at the end of a data packet, which uses a calculation based upon treating the packet as a number. This number has a division applied to it and the remainder from the division is used as the CRC.

Theory into practice

An ethernet frame (packet) typically has a size of 1542 bytes, with 4 bytes for the error checking. Use a spreadsheet to calculate the largest number that can be held in 4 bytes.

What are the advantages of using CRC over a simple checksum to calculate the error check?

The concepts, implications and applications of error detection

Every time data is transmitted or stored there is a risk of data **corruption**, so error detection is built into computer systems and protocols are set up to try to eliminate these problems. Generally this involves adding numbers which are mathematically related to the original data (such as a checksum or CRC) so they can been recalculated at the receiving end. However, calculating and recalculating these values adds to the processing overhead. These error detection schemes also decrease the real data transmission rate. They involve sending additional data which is not part of the actual data being transmitted: once at the receiving end, this error detection data is checked and discarded. Thus the overall efficiency of the data transmission system is reduced.

It is probably impossible to ever be certain of finding 100 per cent of any errors as there will always be a possibility that separate corruptions could 'correct' an error that has been detected.

Key term

Corruption – in the context of data, is when one or more bits get misunderstood by the receiving computer system during a data transmission.

PAUSE POINT The cause of a corruption in data varies for different media, as the corruption has to affect the signal inside the medium. For example, it could be caused by an electrical cable being placed close to a network cable. What are the possible causes of corruption for different media types?

Hint What is the signal used to transmit data in (a) cat 6 copper cable (b) fibre optical cable (c) WiFi and (d) laser point-to-point network connections?

Extend What would cause a corruption for each of these transmission methods? Which transmission method is the most reliable and why?

Error correction

Once errors have been detected they need to be corrected, and there are a number of error correction methods.

Commonly used error correction systems

This section looks at two commonly used error corrections systems – automatic repeat request and forward error correction.

Automatic repeat request (ARQ)

The ARQ method of responding to detected errors is to request a retransmission from the sending computer system. Any bad packets (those which contained errors) that were received are not used and are replaced by the resent data.

The protocol controlling data transmission may require the receiver to confirm that data packets have been successfully received, with the sender retransmitting any packets that have been identified as having errors or that have not been acknowledged by the receiver.

Forward error correction (FEC)

FEC is a technique whereby extra data is sent with the transmission so that the receiving system can rebuild bad data. There are sophisticated techniques used to implement FEC that involve repetition and extra parity bits.

Link

Look back at the section on 'Error detection' for more about parity and repetition schemes for error detection.

The concepts, implications and applications of error correction systems

Error correction systems are put in place to respond to any problems in the logical accuracy of transmitted data and to respond by retransmitting any bad data.

Error correction can only be as good as the detection systems used to identify errors in the first place.

Assessment practice 2.5

A01

Use a seven letter word to demonstrate how to use a Vigenère cipher to encode a word. Write a brief description of how you encrypted the word.

Produce a guide to methods used by computer systems to detect errors from data transmissions.

You should:

- explain how a computer system can use even parity checks
- describe and compare how ARQ and FEC are used in error correction systems.

Plan

- How will I approach the task?
- Are there any areas I think I may struggle with?
- How confident do I feel in my own abilities to complete this task?

Do

- I understand my thought process and why I have decided to approach the task in a particular way. I can explain this reasoning when asked.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.
- What am I struggling with? Do I know how to overcome this?

Review

- I can explain how I approached the task.
- I can explain which elements I found easiest.
- I can explain which elements I found hardest.

F The use of logic and data flow in computer systems

All the circuits inside a computer system use digital logic. This section explains the uses, applications and interpretation of logical processes. Diagrams are used to represent the data flow and relationships in and between computer systems to help communicate their relationships and workings.

Boolean logic

Boolean logic is based on logic gates. These are tiny electronic circuits that can accept input(s) and produce an output according to the logic of the input(s). There are hundreds of millions of these circuits inside components such as the RAM, CPU and other chips used in a computer system.

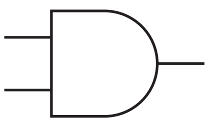
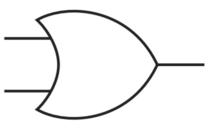
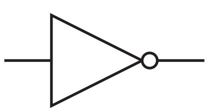
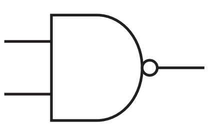
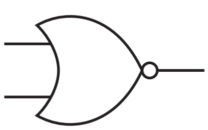
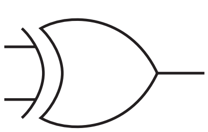
<table>
<tr><th>Datename</th><th>Symbol</th><th>Notification</th><th>Truth table</th></tr>
<tr><td>AND</td><td></td><td>F = A∧B or
F = A.B</td><td><table><tr><th>A</th><th>B</th><th>AB</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>OR</td><td></td><td>F = A+B</td><td><table><tr><th>A</th><th>B</th><th>A+B</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>NOT</td><td></td><td>F = $\overline{A}$</td><td><table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>NAND</td><td></td><td>A = $\overline{A.B}$</td><td><table><tr><th>A</th><th>B</th><th>AB</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>NOR</td><td></td><td>F = $\overline{A+B}$</td><td><table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>XOR</td><td></td><td>F = A⊕B
F = A≠B
F = $\overline{A}B+A\overline{B}$</td><td><table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
</table>

Figure 2.18: Truth tables

Truth tables

The outputs from a logic gate can be shown using a truth table. All the possible input combinations are shown to the left of the truth table as 0s and 1s, arranged in a binary numerical sequence. An output of 0 or 1 is shown for each input combination (see Figure 2.18) according to the logic of the gate.

- AND gate has two or more inputs producing 1 as output when all inputs are 1.
- OR gate has two or more inputs producing 1 as output when any of the inputs are 1.
- NOT gate has one input which is inverted as the output.
- NAND gate (AND with NOT) has two or more inputs producing 0 as output when all inputs are 1.
- NOR gate (OR with NOT) has two or more inputs producing 1 as output when all inputs are 0.
- EXCLUSIVE OR (XOR, EOR) gate has two inputs producing 1 as output when the inputs are different.

The use, application and interpretation of Boolean logic to identify data flow and solve problems

The starting point for using Boolean logic to identify data flow and solve problems is to produce a truth table showing all the inputs into the circuit and all of the outputs. Every output where a 1 is needed has the input combination defined as a Boolean term. These Boolean term definitions can then be used to design the logical circuit.

Inside the CPU is the ALU (arithmetic and logic unit) where calculations are worked out. An example of a simple calculation is to add two numbers together. These numbers will be inside the ALU as binary bytes, simplified into 8 BITs, as shown in Figure 2.19.

	128	64	32	16	8	4	2	1	
	0	0	1	1	0	0	1	1	A (51)
+	0	0	0	1	0	1	1	1	B (23)
	0	1	0	0	1	0	1	0	Sum (84)
	0	0	1	0	0	1	1		Carry

Figure 2.19: Binary addition

Notice that the first calculation in the 1 column adds together the least significant digits from A and B with no carrying involved, as all the other calculations need to include the number carried from the previous pair of digits. The circuit used for this is called a **half adder (**see Figure 2.20: Half adder).

The truth table for the first calculation is shown in Table 2.6.

Table 2.6: Half adder truth table

A	B	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

The carry only has 1 when both A and B are 1, so can be defined as:

Carry = A AND B

The sum only has 1s when A and B are different, so can be defined as:

Carry = A XOR B

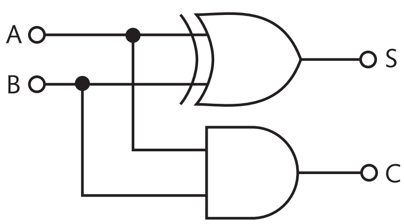


Figure 2.20: Half adder

Notice that the rest of the calculations in Figure 2.20 add together digits from A and B and the number carried from the previous pair of digits. The circuit used for this is called a **full adder**. The truth table for this is shown in Table 2.7.

Table 2.7: Full adder truth table

A	B	Carry in (Cin)	Carry out (Cout)	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

The carry out is 1 for several combinations of A and B that have carry in equal to 1, so needs to be defined as a term for each 1, where the terms have a logical OR relationship.

Each term uses AND between every input with any input at 0 inverted to make it into 1 for the AND operation to produce the 1 output.

Boolean algebra simplifies the writing of the expression by using a line over a term for NOT, a full stop for AND and a plus for OR:

$$\text{Carry out} = A.B.Cin + A.B.Cin + A.B.Cin + A.B.Cin$$

This expression needs to be simplified before a logical circuit can be designed to implement the carry out part of the half adder. One method is to use a **Karnaugh map** to reduce the terms.

A Karnaugh map is a grid with the possible inputs around the outside and outputs indicated using a symbol such as x. The inputs are arranged so that just one part of a term is different from the one next to it, as shown in Figure 2.21.

	$A.B$	$\overline{A}.B$	$\overline{A.B}$	$A.\overline{B}$
Cin	x	x		x
$\overline{C}in$	x			

Figure 2.21: Karnaugh map for Carry out

From the Karnaugh map, we can see in the first column that Cin makes no difference to A.B, so we now have:

$$\text{Carry out} = A.B + \overline{A}.B.Cin + A.\overline{B}.Cin$$

This can be simplified to:

$$\text{Carry out} = A.B + Cin\ (\overline{A}.B + A.\overline{B})$$

Because $\overline{A}.B + A.\overline{B}$ is the Boolean expression for an exclusive OR gate, the carry out circuit needs to OR together the result from A AND B (A.B) with the result from Cin AND the exclusive OR of A with B.

$$\text{Sum} = \overline{A.B}.Cin + \overline{A}.B.\overline{C}in + A.\overline{B}.\overline{C}in + A.B.Cin$$

$$\text{Sum} = Cin(\overline{A.B} + A.B) + \overline{C}in(\overline{A}.B. + A.\overline{B})$$

$$\text{Sum} = Cin(\overline{A}\oplus\overline{B}) + \overline{C}in(A\oplus B)$$

$$\text{Sum} = Cin\oplus A\oplus B$$

We now have the Boolean expressions needed for a full adder circuit (see Figure 2.22).

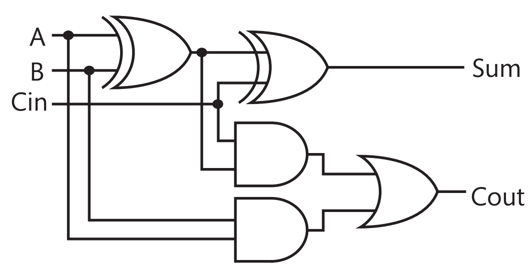


Figure 2.22: Full adder

The use, application and interpretation of Boolean logic to identify logical structures, represent data flow and solve problems

Hardware designers use Boolean logic to help them to identify logical structures which can then be made into circuits. In the early days of computing, these circuits were

designed using drawing boards but, during the last few decades, computers have been used to make the design of tiny complex logical circuits possible.

Boolean logic can be used to represent data flow and solve problems with logical structures. The inputs into and outputs from a circuit can be defined in a truth table that has a column for each input and each output, and where every output is used to define a term in the Boolean expression for that column. As already seen, these expressions can be simplified to give the final solutions.

Flow charts and system diagrams

There are many ways in which diagrams are used to help explain how a computer system works.

Flow charts

Flow charts and other diagrams use arrows to represent data flow, where the direction is shown by the arrow. Flow charts also use a collection of shapes that are used to identify the functions of different parts of the system.

Link

Read the section on 'Flowchart and use of standard symbol conventions', in *Unit 4: Software Design and Development Project* to learn about the different shapes and arrows that are used in flow charts.

Flow charts are an important tool for both design and debugging as they can be used to trace through the expected paths that an algorithm should take with the help of a **dry run** table. This checks that variables are set and changed in the correct way at the right moments.

Key term

Dry run – this is a method used by code developers to carefully follow how a program will run, by keeping track of the contents of variables at every point and checking that the decisions are based upon the correct values so that the program follows the expected code branches.

System diagrams

System diagrams are less formal than flow charts. They use boxes to represent parts of the system and arrows to show the data flows. Flow charts are well suited to program code whereas system diagrams are good for showing how larger systems operate.

Solving problems

Flow charts and system diagrams are both used to explain how systems work and to help solve problems.

A diagram helps to visualise a system and to communicate its inner workings to the designers, developers and other stakeholders of a project. This helps them to understand the system and to discuss any part that is in need of development or repair.

Identified problems within a system can be resolved using these methods, as the circumstances leading up to the issue can be traced from the beginning through the diagram. This path through the system can be used to see where the issue arose, so helping to identify a solution to the problem.

PAUSE POINT What are the real benefits, if any, of using diagrams to help explain how a system works?

Hint Close the book and discuss with a peer how a diagram can communicate in ways that are difficult for words alone.

Extend A program flow chart has standard symbols to represent different types of code. What are the benefits of using standard shapes in a diagram? Who does this benefit the most?

Assessment practice 2.6

A01 A02 A05

Describe and explain the logic circuits needed to add together two 8-bit bytes in a processor. Include truth tables to define the inputs and outputs for each type of circuit used here.

Produce a guide to the symbols and describe the code represented by terminators, arrows, input/output, processes and decisions in a program flowchart.

Produce an example flowchart to illustrate a programming algorithm of your choice with a written description of the algorithm.

Plan

- What am I learning? Why is this important?
- How will I approach the task?
- Do I have any existing knowledge of the task?

Do

- Am I confident that I know what I am doing and that I know what it is I should be achieving?
- What strategies am I employing? Are these right for the task? Are they working? If not, what do I need to do to change this?
- I understand my thought process and why I have decided to approach the task in a particular way. I can explain this reasoning when asked.

Review

- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).
- I can explain where I learn best.
- I can explain what I have learned and why it is important.

THINK FUTURE

Tamas Diossy

IT Support Manager for a major bank

I've been working here for almost nine years. I got an internal promotion to my present position two years ago. My team consists of four technicians who support the IT system of this site of 400 users. The IT support we supply covers hardware, software, networking, purchasing recommendations and approvals. I like the scope and variety of the job, although I do miss programming as this skill is not required in my present role.

Any purchasing of new IT equipment or software needs to be approved by my department. We have a good understanding of the current marketplace and our current IT systems and can usually make one or two recommendations to any department needing new equipment or an upgrade to their system. We need to approve their decision as part of the sign-off process for the purchase. This process works well, we are pleased with the how the current IT system performs and we do our best to keep it that way by ensuring that new equipment is compatible with the current system, has a good performance rating and is good value for money.

Focusing your skills

Computer hardware within a computer system

Many job roles require an understanding of the current technologies and choices that can be made when purchasing new IT equipment as replacements or upgrades. Here are some suggestions for how you can improve your employability by improving your expertise and knowledge in this area.

- Understand the factors affecting the choice, use and performance of internal computer components. Be particularly aware of the need to get good value for money by choosing equipment that meets the needs of users without adding unnecessary extra functionality or performance, which adds to the cost.
- Be clear about how meeting needs includes providing an acceptable user experience, ease of use, high performance, availability, accessibility, compatibility, being within budget, availability within an acceptable timescale after testing/migration to a new system, productivity and security.
- Build up knowledge of the current options for data storage and recovery systems including RAID and NAS as well as announcements of new technologies and products.

The concepts of microarchitecture

Processors are one of the main components in a computer system. Knowing the concepts of microarchitecture will help you to understand and apply CPU specifications and features to help you to make an informed choice when purchasing a new processor and motherboard.

- Be clear about the factors affecting execution speeds, methods of achieving faster execution and implications of faster speeds. You also need to know the amount of processing power needed for different software applications.
- Find out the role of the cache inside a CPU. Research current processors to get to know how much cache is normal, small or impressive.
- Find out the role of cores inside a CPU and the benefits of more than one core. Research current processors to get to know how many cores are normal, small or impressive.
- Regularly use an independent website such as Tom's Hardware to see the results of testing hardware to build up your knowledge of how well different items of hardware perform and how much extra performance can be expected from the various performance features.

Getting ready for assessment

This section has been written to help you to do your best when you take the assessment test. Read through it carefully and ask your tutor if there is anything you are still not sure about.

About the test

The assessment test will last 1 hour and 45 minutes and there are a maximum of 80 marks available. There are 3 types of question in the test:

- annotated diagram questions
- short answer questions worth 1–2 marks
- a longer answer question worth up to 12 marks.

Remember that all the questions are compulsory and you should attempt to answer each one.

Each question will state how many marks are on offer for its completion. You should attempt all the questions.

Sitting the test

Arrive in good time so you are not in a panic.

The test will be carried out under supervised conditions.

- You may use a calculator.
- Internet access is not permitted.
- You must not bring anything into the supervised environment or take anything out without your tutor's knowledge and approval.

You will not be allowed to take your mobile phone into the exam.

Make sure that you arrive in good time for each test session and check that you have everything you need for the test ahead of time.

Listen to, and read carefully, any instructions you are given. Lots of marks are lost through not reading instructions properly and misunderstanding what you are being asked to do.

Organise your time, based on the marks available for each question. Set yourself a timetable for working through the test and then stick to it – do not spend ages on a short 1–5 mark question and then find you only have a few minutes for a longer 7–8 mark question.

Remember that you cannot lose marks for a wrong answer, but you cannot gain any marks for a blank space!

Ensure that you check all sides of the assessment task before starting work.

Ensure that you leave yourself enough time to check through your work. Proofread and correct any mistakes before handing in your work

Command words typically used in assessment

Most questions contain command words. Understanding what these words mean will help you understand what the question is asking you to do.

Command word	Definition – what it is asking you to do
Analyse	Identify several relevant facts of a topic, demonstrate how they are linked and then explain the importance of each, often in relation to the other facts.
Calculate	Apply some form of mathematical or computation process.
Complete	Complete a diagram or process. This can be applied to problems/solutions of varying complexity.
Demonstrate	Illustrate and explain how an identified computer system or process functions. This may require a piece of extended writing, drawing a diagram or a combination of the two.
Describe	Give an account of something or highlight a number of key features of a given topic. This may also be in relation to the stages of a process.
Develop	Provide a solution to a problem, typically using an existing system or structure that must be improved or refined.
Discuss	Investigate a problem or scenario, showing reasoning or argument.
Draw	Represent understanding through the use of a diagram or flow chart.
Evaluate	Review and synthesise information to provide a supported judgement about the topic or problem. Typically, a conclusion will be required.
Explain	Make a series of linked points and/or justify or expand on an identified point.
Identify	Assess factual information, typically when making use of given stimuli. Requires a single word or short sentence answer.
Produce	Provide a solution that applies established constructs to a given computing problem.
Write	Produce a solution, or a mechanism used as part of a solution, to a given computing problem.

Writing long answers

If you are writing a longer answer, try and plan before you start writing. Have a clear idea of the point your answer is making, and make sure that this comes across in everything you write, so that it is all focused on answering the question.

- Always make a plan for your answer before you start writing. Sketch this out so that you can refer to it throughout – remember to include an introduction and, if the command word is ‘evaluate’, a conclusion and think about the key points you want to mention in your answer. On this plan, set yourself some timeframes so that you make sure that you have time to cover everything you want to and, importantly, have time to write the conclusion!
- Try to keep your answer as focused on your key points as possible. If you find your answer drifting away from that main point, refer back to your plan.
- Make sure that you understand everything being asked of you in the activity instructions. It might help you to underline or highlight the key terms in the instructions so that you can be sure that your answer is clear and focused on exactly what you have been asked to do.

Sample answers

For some of the questions, you will be given some background information on which the questions are based.

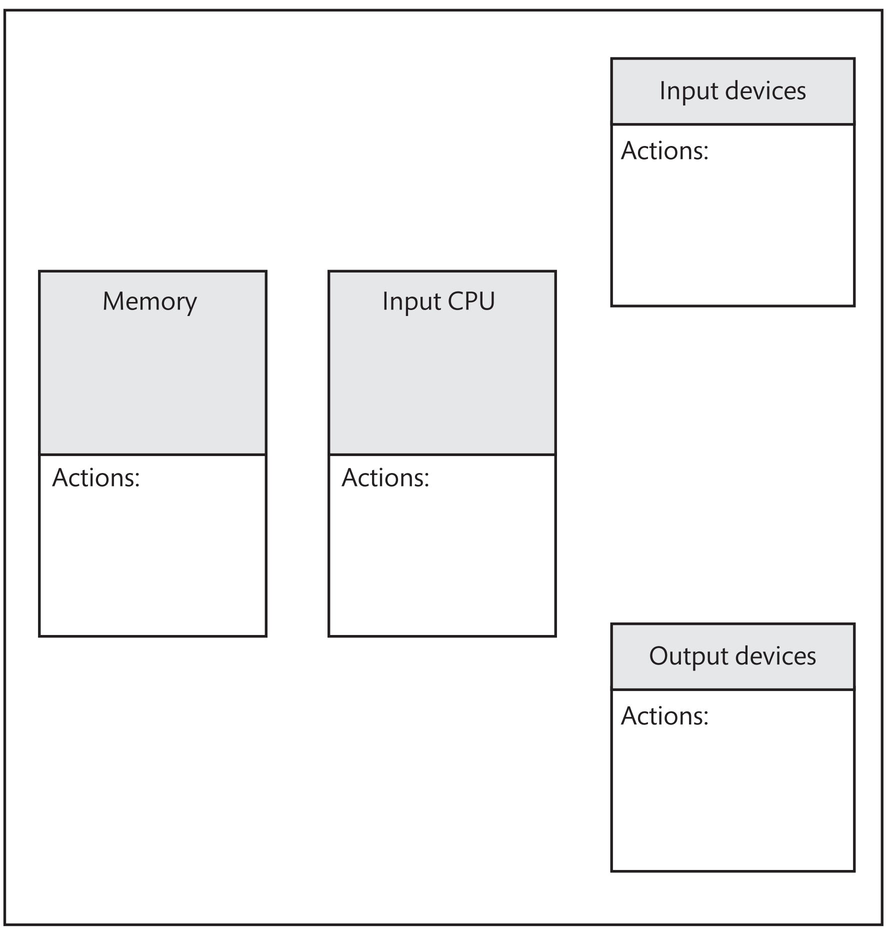
Look at the sample questions which follow and our tips on how to answer these well.

Answering annotated diagram questions

You need to be careful when annotating a diagram. There will be a small amount of space so carefully choose words which are suitable for the **context** of the question. Always read the question carefully and write the **most appropriate** answer.

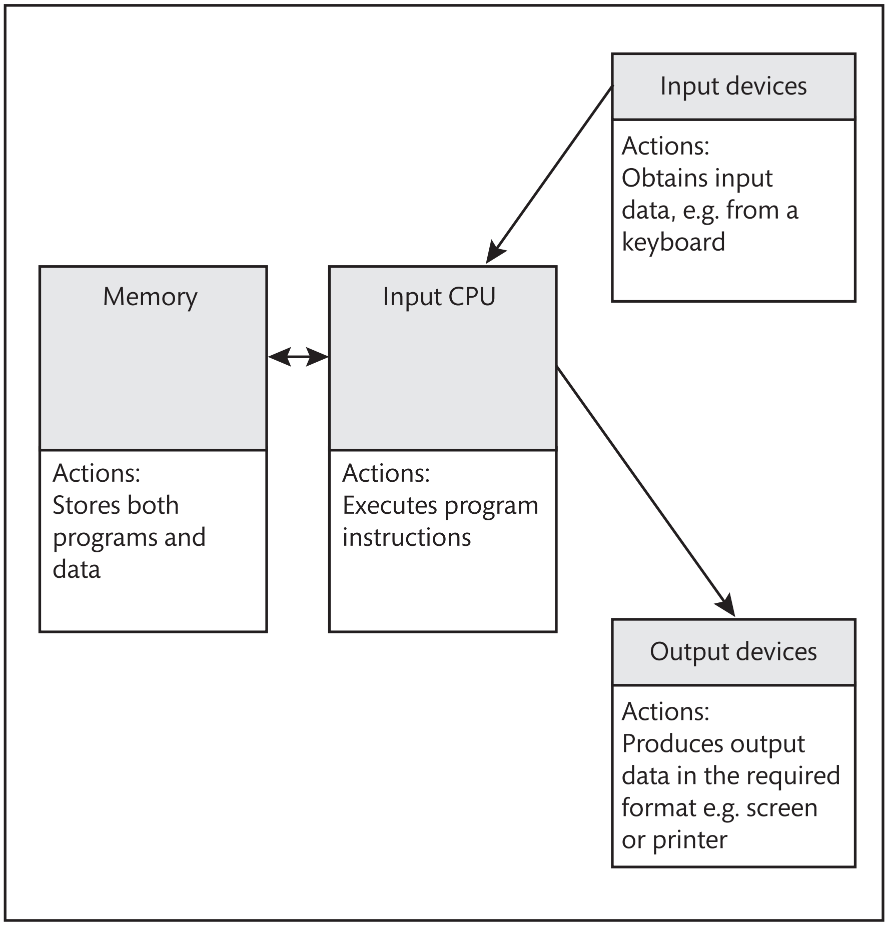
Worked example

Many computers use the Von Neumann architecture for their CPUs.
The main components of the Von Neumann architecture are show below.
Compete the diagram by describing the actions performed by each component and the direction(s) in which data flows:



To get full marks the data flow arrows must point in the correct direction.

Answer:



The examiner will accept any other relevant phrasing/wording alternatives to the model answer given here.

Answering short answer questions

- ☐ Read the question carefully.
- ☐ Highlight or underline key words.
- ☐ Note the number of marks available.

Make sure that you make the same number of statements as there are marks available. For example, a 2 mark question needs 2 statements.

Look carefully at how the question is set out to see how many points need to be included in your answer.

Worked example

State two types of operating system than can be used on a computer **[2]**

Answer: Real-time, multi-tasking.

This answer is enough to achieve the 2 marks. Other possible answers include single user and multi-user.

Answering extended answer questions

Worked example

Analyse how RAID discs can be used by server computers **[6]**

For a question using the word 'analyse', you must identify several relevant facts of the topic, demonstrate how they are linked and then explain the importance of each, in relation to the other facts.

Answer: Redundant Arrays of Independent Discs (RAID) is a disc storage technology that has two main benefits, improved performance and data security. Because the data is spread across at least two different discs this provides better performance because there are multiple drives reading and writing data rather than just one large drive. This is important for server computers because they usually support large number of users, all of which may be requesting data off the discs at the same time. Using RAID drives with multiple discs increases the number of simultaneous reads/write and therefore provides better performance for users.

Make sure you state not just how the RAID system works but also why it is important in a server computer environment

RAID also provides fault tolerance by spreading data across multiple disc drives using a technique called striping and by storing parity data across the drives. The means that if one drive in the RAID set fails the data that was on the drive can be calculated using the parity data. The most commonly used versions of RAID, RAID 5 and 6 allow a server computer to continue uninterrupted if a single drive fails. Once the failed drive is replaced the data on it is reconstructed. Fault tolerance is important for many servers because they often support business critical applications which would cause considerable business disruption if they were not available.There are no moving parts as the SSD uses flash storage which reduces the overall weight of the tablet making it more portable. There is no mechanical operation, which makes it silent and lets the SSD run without much heat generation.

Planning and Management of Computing Projects 3

Getting to know your unit

Assessment
You will be externally assessed by a two part test.

Every day of our lives involves some form of project planning, whether it is planning a meal, a holiday or preparing for college. In this unit, you will study how project planning and management concepts are applied to computing projects. Computing projects feature in every industry, regardless of size or type, and each one requires careful and systematic planning to execute a successful outcome.

This unit explores the business case needed for the initial approval of a computing solution to meet organisational needs. Learning about designing and monitoring projects will provide you with the skills associated with project planning and management. Good planning and management skills are essential to ensure that an end product can be delivered on time, within budget and to the required specification. These skills include task scheduling, budgeting, risk management, time management, quality management and communication with all stakeholders throughout the life cycle of the project.

In this unit, you will apply project planning and management techniques to a computing project scenario. This will develop your knowledge and understanding of the role of a computing project management professional and support your progression to higher education studies.

How you will be assessed

The set task will be completed under supervised conditions in two sessions. Part A should last three hours and Part B should last two hours.

The set task will assess your ability to plan and manage a computing project.

Grade descriptors

To achieve a grade, you will be expected to demonstrate these attributes across the essential content of the unit. The principle of best fit will apply in awarding grades. The maximum number of marks for this unit is 66.

To pass this unit:

- You will be able to use your knowledge of project planning, management concepts and processes and the application of problem-solving skills to show the documenting of project planning and management requirements. These are limited in scope and may be incomplete.
- You will be able to use planning and management documentation, and demonstrate an understanding of your completion and development to a minimal level of acceptability in order to support an organisation's project. Your evaluation of a given project planning and management scenario is limited in scope and may be incomplete.

To gain a level 3 distinction:

- You will demonstrate that you can evaluate a given project planning and management problem and develop a detailed and complex project planning and management documented solution to effectively meet all project scenario requirements.
- You will demonstrate an in-depth understanding of project planning and management documentation requirements and are able to show that you fully understand how these are used to produce an effective project solution.
- You are able to evaluate your solution in order to make justified recommendations on project development and future actions.

Assessment outcomes

AO1 Demonstrate knowledge and understanding of the project planning and management concepts, processes and life cycle.

AO2 Apply knowledge and understanding of computing management tools, techniques and procedures to explore outcomes and find solutions to problems.

AO3 Analyse data and information, recognise patterns, correlations and connections in order to solve problems and predict outcomes.

AO4 Evaluate project planning and management tools, techniques, procedures, outcomes and solutions to make reasoned judgements and make decisions.

AO5 Be able to plan a computing project and manage it throughout its life cycle, with appropriate justification.

Getting started

A project is a one-off, temporary task that produces an end result as opposed to an ongoing programme of events. As technology becomes ever more sophisticated, computing projects of all sizes are undertaken daily in business. Provide examples of where you have experience or knowledge of each stage of project management.

A Project management concepts

The purpose of project management is to plan, organise, budget and control the project so it meets the needs of the customer within the constraints given. Project management provides a framework which is broken down into objectives and designed to meet the overall aim of the project. In this section you will explore the key factors, processes and stages that make up a typical computing project.

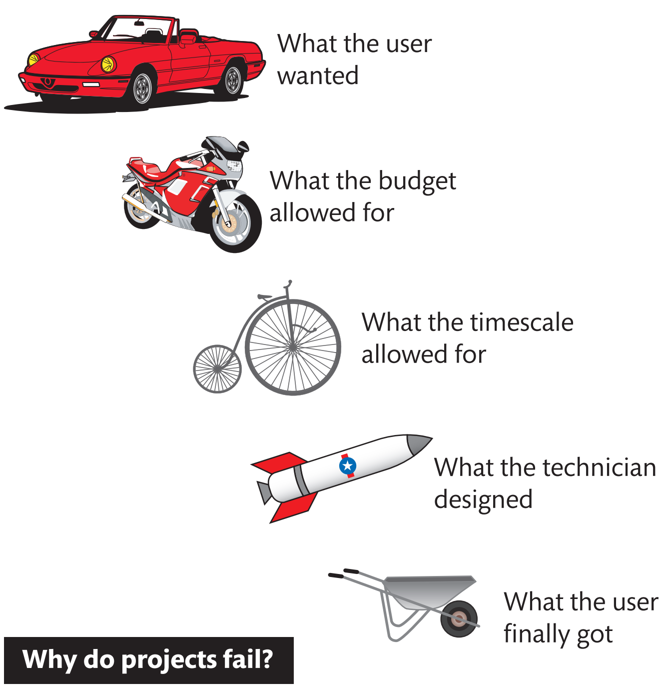


Figure 3.1: Why do you think this situation arose? How could the outcome have been prevented?

Costs and timescales

Every project has a budget and a timescale for completion just as you do when you make a purchase of a product or service. Every aspect of the project must be considered even if there does not appear to be a direct cost, such as purchasing parts or paying for **sub-contractors**.

Travel expenses, the time taken by anyone on the project through phone calls and administration, all need costing into a project otherwise you are at risk of the project not being viable. In other words, it will be likely to make a loss as opposed to the project being successful (viable). One of the difficulties faced by project managers is estimating as accurately as possible the amount of time each aspect of the project will take, and therefore the cost incurred. The actual cost of the project when completed is used to calculate your profit so your estimates need to be accurate otherwise you will face unexpected expense and might find that the project was not worth the investment.

The following section will look at how three key factors can be used to determine project viability and measure progress and success. These key factors are project budget, setting milestones and deadlines, and interim reviews.

Key term

Sub-contractor – an individual or business paid to work on a project, usually known as a third party, and not an employee of the company that the project is being carried out for.

Project budget

A project budget is a sum of money allocated to undertake and complete a task, in this case computer-related. It is the amount of money that has been set aside for the entirety of the project from start to end. It will be broken down into each individual expected cost.

Businesses often underestimate project budgets, possibly due to limited knowledge of the complexities and labour

involved or perhaps just wishful thinking. The project manager (PM) is responsible for planning the project and overseeing its execution within the constraints of the budget. If the project is unlikely to be completed within the budget stated at the outset then the PM must be honest with the customer and be able to justify why or provide alternative suggestions to keep within budget. It is important for the client to also be honest with the PM to develop a trusting relationship. It is possible that the client may have unrealistic expectations, often due to lack of understanding of costs and complexities, or they might want or need to change the scope of a project part way through, in which case it is unreasonable to expect the budget to remain the same.

Often projects specify a **contingency** fund to allow for unexpected expenditure incurred by, for example:

- rising costs of labour
- additional time needed due to unexpected occurrences, such as delays in supplies, or resources being needed for other projects running alongside
- the need for additional resources such as materials or staff.

Key term

Contingency – anticipated allowance made for unexpected future events, in this case additional funds for a project.

Setting milestones and deadlines

In order to monitor and measure the progress of any project, the PM will set milestones and deadlines for achieving different stages. These milestones are like those you agree when action planning, perhaps with a personal tutor, to monitor and measure your progress throughout your studies.

Each milestone will be set in advance within a timeframe that is agreed by the team as being manageable and reasonable for the project's progress. By setting these milestones, the PM can organise resources (people, materials and equipment) in accordance with each stage and keep better control over the budget by buying what is necessary sufficiently in advance without buying it in too early.

Every project must have a **schedule of works** which enables planning of the project against the deadlines set by the client. The client will have reasons for the deadline they have set but, when discussed with the PM, this deadline may require some adjustment before the project is embarked upon, perhaps due to the availability of materials and other resources. A deadline might also be adjusted due to the complexity of the project, because it may not be feasible in the original timeframe allowed for by the original deadline.

Key term

Schedule of works – a plan or procedure defining the activities that need to be carried out to achieve the project objectives. It is sometimes referred to as a statement of work.

Link

For more about schedule of works see the section 'Objectives, written as SMART targets'.

Once agreed, the client will expect the project to be delivered on time, by the deadline. A consequence of not meeting the agreed deadline may be penalty payments being incurred by the PM. (Obviously this depends on the terms of the contract the PM agreed with the client.) Therefore the PM should always consider, before agreeing the overall budget with the client, whether or not there is a risk of not meeting the milestones or deadline.

Link

How PMs manage risk is covered later in this unit, in the section 'Risk'.

Research

Find out more about penalty payments and fixed price contracts. You could start by looking at these examples.

- Penalty payments: https://www.gov.uk/government/speeches/street-works-overrun-penalties.
- Fixed price contract project: http://www.computing.co.uk/ctg/news/1845773/nhs-project-cost-gbp124bn which may avoid penalty payments.

Interim reviews

The PM breaks the project down into smaller chunks (phases of development/tasks). This enables the team to organise the work involved in each step in sequence, and makes it possible to assess whether or not the project is progressing according to the milestones at interim reviews.

Each member of the team should be involved in interim reviews at the end of each stage of development, to provide a comprehensive review of progress and so they can discuss any difficulties encountered or changes that are needed to the overall project plan.

Quality and deliverables

Successful project implementation relies on frequent monitoring, testing and planned **quality** checks. Quality frameworks are designed to support quality checks of both hardware systems and software products against the objectives and required deliverables of a project.

Deliverables are the agreed outcomes of the project. For example, a client who wants a fully-integrated computer network for staff to access customer records when working remotely will expect a fully operational system as the main deliverable. The quantifiable components which lead to this end product are also deliverables, for example cabling, connectivity and a customer database. Each of these deliverables will be quality checked throughout the development process.

Quality checking relies on gathering data to measure outputs against the project requirements. For example, a series of IT network test results can be analysed and the problems can then be identified and rectified. You may have carried out some quality testing of code that you have written. By checking the code, you can recognise, for example, where gaps in the code interrupt the execution of a program.

Application of current quality standards and subsequent iterations

There are two types of quality standards for software development and website design, and these are discussed in this section. Quality processes usually have several **iterations** and are carried out over a predetermined timeframe. This enables layers of checks to be made so that comparisons between stages can be used to identify opportunities for improvement.

ISO/IEC 25010:2011 as a benchmark for software development

ISO/**IEC** 25010:2011 is a quality standard which has replaced the previous 9126 version to reflect current expectations and technological developments. Its purpose is to provide a standardised framework for measuring and assuring the quality of the development. It is recognised internationally by developers and customers alike. All parties are likely to have an expectation of the quality of the finished article, and these expectations are managed by agreeing on a **benchmark** to be met or exceeded. This benchmark might be the minimum acceptable standard expected.

The application of this standard includes systematic checks, carried out from inception and iteratively throughout the project, that involve:

1. identifying software and system requirements
2. validating the comprehensiveness of a requirements definition
3. identifying software and system design objectives
4. identifying software and system testing objectives
5. identifying quality control criteria as part of quality assurance
6. identifying acceptance criteria for a software product and/or software-intensive computer system
7. establishing measures of quality characteristics in support of these activities.

Key terms

Quality – this is the term given to a standard of measurement that can be graded against similar kinds when compared with each other or with a series of benchmarks.

Iteration – something that is repeated over a cycle or period, often building upon each previous stage.

ISO – Internal Organisation for Standardisation.

IEC – International Electrotechnical Commission.

Benchmark – a point of reference that provides a measurement against which a comparison can be made.

ISO 9000 framework

You may have heard of ISO 9000 or one of its series, and perhaps seen it promoted on a company's website or in its advertising literature, for example Moneycorp and some government departments. ISO 9000 and similar frameworks are sets of standards for managing quality by providing guidelines and procedures which businesses use to check that deliverables and processes meet prescribed requirements. You might associate these standards with those that qualifications, such as this one, use to produce specifications in order to ensure that you are taught what industry requires.

These standards are recognised widely and, although not compulsory, some companies will only do business with other organisations that are ISO accredited because this gives them confidence in their procedures and quality checks.

World Wide Web Consortium (W3C®) for website design and functionality standards

In 1994 Tim Berners-Lee founded the W3C to initiate universal codes of practice for accessibility of the internet. The WCAG defines guidance for creating web content that avoids discrimination by providing three basic tips on:

- user interface and visual design
- writing and presenting content
- mark up and coding.

Link

For more about W3C, visit http://www.w3.org/

Link

For more about mark up and coding, see *Unit 1: Principles of Computer Science* and *Unit 4: Software Design and Development Project*.

Further research and technological development led to more advanced ways of sharing information, resulting in Web 2.0 just after the turn of the century.

As the use of the internet continues to grow and develop, W3C continues to strive to eradicate inaccessibility, for example, by defining terminology, user needs, providing guidelines and resources for use and designing for inclusion, such as mobile accessibility and content.

Link

For more about W3C on mobile accessibility and content, visit http://www.w3.org/WAI/mobile/

Defining success criteria and using SMART objectives to define project outcomes

In order to plan and manage a project successfully, the overall aim is translated into deliverables that form the success criteria, in a similar way to how your qualification is structured. This enables the PM to plan the project in stages and identify key performance indicators (KPIs) for measuring progress against the milestones.

Each of the success criteria comprises objectives that define the project outcomes. These objectives might identify the stages for networking a new IT system, developing a bespoke database or building a website. Whatever the objectives, they need to be SMART: meaning that they need to be **S**pecific, **M**easurable, **A**chievable, **R**ealistic and **T**imebound.

- Specific: The objective needs to be clearly stated and should say exactly what it is. Therefore if an objective is too broad, such as 'network the offices', this is classed as an aim and should be broken down into manageable (specific) portions which are the objectives such as 'complete cabling' or 'install switches'.
- Measurable: Every objective needs to be measurable to enable an assessment of whether it has been executed according to the requirements. Each objective will need quantifying to be measurable by stating what will be achieved as a result of the objective, for example a measurement of increased productivity or a 10% increase in monthly production output.
- Achievable: An objective needs to achievable and therefore realistic against the parameters given. This might mean that the plan of how to meet an objective might need rethinking if it cannot be achieved within those parameters. For example, an objective might appear achievable but, when evaluated in terms of quantifying its intended outcome or output, it might not be achievable in practice, such as a helicopter flying around the world without stopping.
- Relevant: It is possible for an initial objective to simply not be relevant to the success of the project. As each objective is evaluated, this evaluation also follows an iterative process by assessing each objective against each of the SMART objectives. For example, a project to implement an accounting programme is unlikely to include additional software for desktop publishing, unless explicitly requested by the client in their brief so any such objective would not be relevant and should be removed because it is not SMART.
- Timebound: Every objective must have a specified deadline, just as your assignments and action plans have time constraints relating to them. There are time constraints on every project and, as you may be aware, there are numerous news articles about major projects that do not run according to schedule.

An example of a SMART target in an office networking project might be: 'All 4 routers to be installed, tested and working by 5pm 19/9/2018'.

Link

For an example of a project which did not run to time, find out more about the turn of the century NHS IT project which was eventually abandoned after costing more than £12bn: http://www.theguardian.com/society/2011/sep/22/nhs-it-project-abandoned.

The key to a successful project plan is to consider carefully whether each objective is SMART and, ultimately, how well they contribute to meeting the overall aim of the project when put together.

Customer requirements in terms of functional requirements and non-functional requirements

A project comes about when a client requires something to be manufactured or a service undertaken, for example a publishing company requiring a new software platform or an individual wanting a new kitchen designed and fitted. The client will have preconceptions about the finished deliverables of the project, which will be based on a set of requirements. These requirements are unlikely to have been comprehensively worked out, particularly where the client is not an expert in the area – hence the need for a PM. It is important to remember that the client, rightly, has their expectations and will be paying for the deliverables. Their requirements will be based almost entirely on their expectations as an end user. The PM, although not always a technical specialist in the relevant area, will be relied upon to know (or to find out) what is achievable and what is possible within the budget and timeframe available.

For example, regardless of budget, some things are simply not yet achievable, such as a digital device which accurately tells you the future. However, the PM or their team will be able to advise on alternatives and manage the expectations of their clients.

In terms of a computing project, non-functional requirements comprise those requirements that are devised to assess the operation of a system, whereas the functional requirements are those that are required for the system to operate. The latter are known as behaviours and tell a system what it has to do to meet the needs of the user, such as touch screen operation or a speech recognition trigger. Non-functional requirements exist in the architecture of the system, for example data gathered from usage and security attacks.

Product description or product breakdown structure, to describe the product to be delivered

When managing a project, both parties, that is, the PM and the end user (or contractor on behalf of the end user), should compile a list of what is required and for what purpose. This list is the product description or product breakdown structure. This should be done because both parties will have different ideas (although some will be shared) about what the deliverables are. It is likely that technical jargon that is not explained will result in misunderstandings about what will be delivered by the project.

It is especially important to note that the skill in articulating exactly what is wanted and then interpreting exactly what was meant, by either party, is a significant one because this is the most common area for conflicts to arise. Using a quality framework, such as ISO/IEC 25010, can help to anticipate potential difficulties and provide a checklist for the various meetings that will take place during the life of a project.

Risk

Every project has potential risks attached and those risks need to be anticipated and strategies need to be put in place to reduce or eliminate them. One of the risks has been suggested already, that is, the misunderstandings between communicating parties. Miscommunications could result in a project being cancelled or the PM or a team member being removed from the project. Projects will often include a clause that prevents either party from being able to renege on the contract, although clauses may exist to forestall such failure.

Identifying typical project risks

Earlier in this unit, you were introduced to the notion of risk in relation to project management. Some level of risk is typical with any type of project, for example the risk of interruptions caused by weather conditions on an external construction project. Predictable risks present a form of benchmark and will be more easily anticipated than those which are less typical. However, it is worth mentioning here that sometimes the more obvious or typical risks are harder to identify, for example the risks we take each day when getting out of bed while bleary-eyed or crossing the road while texting or talking with others.

Risks fall into two categories, external and internal.

External risks

External risks to a project are those which are outside of the direct control of the project team.

An external risk might, for example, be the possibility of a cloud server going down, which could interrupt the smooth running of a project and impact on the deadline set. Another example could include emergency works to the broadband connection used by the business, perhaps as a result of upgrading services. These external risks might seem extreme but may have an impact on availability of team members working remotely, access to telecommunications or the delivery of supplies necessary to complete a project.

The project system could be undergoing a series of tests and quality checks when the wireless connection is interrupted or the bandwidth becomes insufficient to execute the checks. Such an incident could significantly affect the smooth operation of a project, and therefore contingency plans should be put in place when planning any project, just as you should when planning for your assignments.

Internal risks

Internal risks to a project are those which are within the control of the project team.

Internal risks include those related directly to the team members, for example illness amongst the project team members who are essential for execution of the project is an internal risk. Other factors might include team members leaving the project, which therefore places pressure on the PM to replace the lost specialist skills and project knowledge that are needed to meet project deadlines.

Other typical examples of internal risks may include interruptions to the project caused by the client, for example where the project team needs to work around access to equipment or premises. If, for example, a computing project involved access to an organisation and required team members to go through security checks, such as when working with prisons, government departments or schools, then delays may be incurred when waiting to receive clearance. Some projects may involve overseas travel whereby passports, insurance, visas and vaccinations may all cause delays.

The risk management cycle

The risk management cycle is a systematic approach to managing risks and establishing ways of mitigating them. There are numerous risk management cycle models and Figure 3.2 is just one example that shows the iterative cycle.

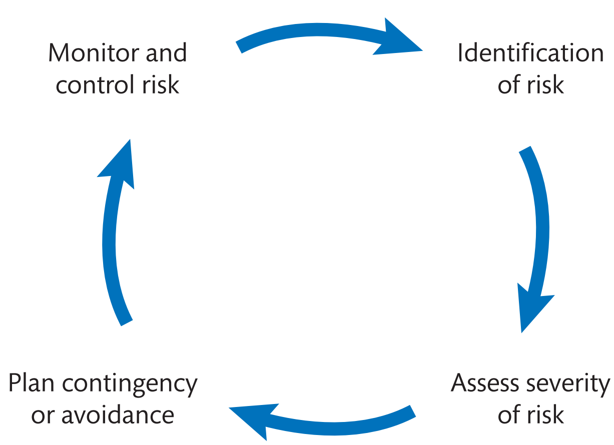


Figure 3.2: The risk management cycle

Identification of risks

Having a frequent and regular monitoring regime in your project plan, including interim reviews, means that you are more likely to identify problems as they occur or even before they arise. These quality checks, such as those identified in the section, Application of current quality standards and subsequent iterations, should each be attached to SMART objectives. The SMART objectives obviously need to be timebound and related to aims or performance indicators. By having frequent quality checks, you or a team member will be able to measure what should be achieved by each milestone and whether the project aims are being met.

For example, systematic testing and analysis of the contents of data-generated reports might indicate interruptions in the system. When investigated, these could reveal problems with bandwidth or perhaps user error, which can then be resolved.

Assessing the severity of risks

Any worthwhile risk management procedure will assess the risks with respect to severity, so that fixing the problems is prioritised. Risks are measured by the degree of possibility that something which has been predetermined will in fact happen. For example in a software development project with a large team of programmers the risk that at least one of the programmers will be off sick for more than one week is quite high. This could delay the project or cause increased costs if a contract programmer needs to be brought in to cover the the programmer who is off sick. There are several methods for assessing the severity of risks, some of which are discussed in this section.

3-point scale for impact and probability

- Using the 3-point scale for impact and probability, the risk factor is measured as a probability and numerically represented. A nil probability (or zero probability) poses very little impact or threat, while a probability of three indicates a high risk. When using a 3-point scale, each risk is given as a percentage and the level of risk can be assessed using a simple range, as shown in Table 3.1.

Table 3.1: Example of 3-point scale for probability

Probability range	Probability value used for calculations	Natural language expression	Numerical score
1% to 33%	17%	Low	1
34% to 67%	50%	Medium	2
68% to 99%	84%	High	3

Link

For more about the 3-point scale for impact and probability, visit https://technet.microsoft.com/en-us/library/cc535373.aspx

Impact multiplied by probability formula

- Once the probability of the risk has been identified, the impact can be measured by applying the probability formula. This formula might represent the impact in financial terms (which is often more meaningful to those involved), in terms of time or by subjective judgements (for example, the impact on staffing or customer satisfaction if perhaps a system is not fully operational).
- In financial terms, each of the 3-point scales will be quantified by a financial loss, differentiated according to the overall value of the project or each individual component of the project, as shown in Table 3.2.

Table 3.2: Example of financial loss attributed to a 3-point scale

Scale	Monetary amount which represents potential loss	Impact on project
1	Less than £500	Budget contingency
2	£500–£2,500	Possible overspend or reduction in completion of full expectations
3	£2,500–£10,000	Project not achievable

Table 3.3 shows the loss in time that the 3-point scale might represent.

Table 3.3: Example of time loss attributed to 3-point scale

Scale	Potential loss of time	Impact on project
1	Less than 1 day	Minimal
2	Up to one week	Significant impact on business
3	More than one month	Project might not be completed

- The risk to the project is calculated by multiplying the probability by the impact. This is shown in Table 3.4.

Table 3.4: Risk of a project calculated by multiplying the probability by the impact

	Low impact = 1	Medium impact = 2	High impact = 3
High probability = 3	3	6	9
Medium probability = 2	2	4	6
Low probability = 1	1	2	3

- The colours and shading identify the high-risk areas (red), medium risk (amber) and low impact (green) by using a traffic light system.

Case study

We have lift off!

When the Soyuz spacecraft left earth on 15 December 2015, British astronaut Tim Peake and his Russian and American colleagues communicated with earth using sophisticated technology enabling sound and vision to be transmitted in real time.

If, for example, the Soyuz spacecraft had not undergone rigorous testing before its launch on 14 December 2015, an undiscovered problem might have existed within the software that enables the astronauts to communicate with people on earth. If this problem had not been revealed until after launch, it would have required those on board to fix the problem so as to avoid even greater problems. After all, they would not have been able to tell anyone on Earth about the problem or learn how to put it right! However, this could appear to be a lower priority risk than if it was found that the rocket was unable to sustain pressure from re-entry through the Earth's atmosphere. Each issue would need to be prioritised appropriately and, although both could carry a significant risk, the time and resources to resolve each problem are likely to be different.

Check your knowledge

1. What is the likely severity of the risk of a complete communications failure?
2. What would be the likely impact of a video communications failure on the arrival and return of the Soyuz spacecraft at the International Space Station?
3. What types of risk could be described as internal and which as external in relation to this case study?
4. What recommendations might you make to mitigate any potential risks to the mission?

PAUSE POINT What other risks can you identify which may be typical of this type of technical connectivity?

Hint Think about the period of time it takes to travel into space to reach the International Space Station in the context of the purpose of the expedition. You can read more about it at https://blogs.nasa.gov/spacestation/2015/09/02/three-soyuz-crew-members-launch-for-two-day-trip-to-station/

Extend Try applying each risk model to assess the level of risk in the scenarios related to the Soyuz spacecraft.

Planning around risks

During the project planning and throughout the quality assurance process of monitoring and reviewing progress during the project, risks will need to be accepted and strategies to mitigate those risks will need to be agreed.

It is highly likely that the level or degree of risk will vary as the project progresses and contingency plans will need to be put in place to mitigate the risks or strategies identified to avoid the risk completely. You cannot completely avoid all risks, as otherwise nothing would ever be achieved, so risks need to be identified and accepted and contingency plans put in place, or the risk (if it does not outweigh the benefit) should be avoided.

For example, if the testing of a new website is scheduled for the same period as when the host server is undergoing maintenance, then the testing is likely to be invalid or incomplete. The host server might incur periods of downtime and therefore the website might not go live. In this example, the contingency plan would be to alter the testing period or to agree with the team maintaining the host server the most suitable time to test your website to avoid the risk as far as possible.

Monitor and control the risks through the project

Throughout any project, it is important to monitor the risks that have been identified as their status will change as the project progresses. For example, if the identified risk was that passengers might not be able to check in online before flying because of a software upgrade, then after the update, the risk might no longer exist.

However, as a result, another risk might emerge such as the software update not being compatible with the other systems in use. Other risks will occur as the project progresses and it is possible that these risks were not anticipated during the initial planning stage. The scenario given above is similar to the problem experienced by Transport for London (TFL) on New Year's Day in 2016 when the fare increase for travellers caused a system failure and passengers with Oyster Cards were able to travel for free.

Handling issues: when a risk occurs and is dealt with using the plan

When something happens that was identified as a risk, such as the problem that TFL had with Oyster Card payments, a solution will be required and quickly. Deciding on how to resolve a problem created by a risk is an essential part of risk management planning. For all known risks, a plan should be created so that, if it does happen, it can be mitigated and resolved quickly. By having a plan you can ensure that the resources to resolve the problem are available when required. The PM should produce a **resource list** to cover all aspects of resourcing the project, including the resources needed to manage risks.

Key term

Resource list – a list of all staff, equipment and raw materials required for a project.

Anticipating the level of risk is necessary for you to identify a contingency plan to resolve the problem. It could be that the level of risk is low but, if that particular problem did occur, the outcome would be significant. Therefore the plan should also identify the priority or severity of the problem should the risk occur. Disaster recovery planning is an example of a contingency plan where the likelihood that fire or flood will destroy computer systems is low but the impact is very high.

Research

Identify a project where a problem occurred. Establish whether the problem had been identified as a risk and how the issue was handled.

An interesting example case study is the Space Shuttle Challenger accident in 1986.

Benefits

This section explores some of the key benefits of a project for an organisation and its stakeholders and how to establish a measurement of success. Any project should

be built around success criteria otherwise it should not be carried out, because there must be a benefit to the organisation. The success criteria will ensure that you know what you are trying to achieve, how you will achieve it and how you will know when you have been successful. In the same way, when you produce an action plan for your studies, it helps you to remain focused and enables you to measure your progress against the milestones you have set.

Business benefits

Businesses expect to benefit from engaging in projects which are well managed and carefully thought through before committing to them. By taking stock of existing business practices and their effectiveness, managers can then gauge how to make improvements and how to maintain those aspects of business practice that they do well. This is a similar process to you gathering all the evidence together and reflecting on your own practices and achievements, for example your studies, personal life and aspirational journey. This process of monitoring, reviewing and evaluation may not necessarily lead to a project of any magnitude but it is important to frequently undergo this process, particularly as it may identify a risk which could have significant impact on future outcomes. Effective project management enables businesses to maximise their profits. For example, an integrated computer system can save considerable time for employees and might also save time for customers too. Take, for example, a restaurant where diners' orders are taken digitally and simultaneously sent to the kitchen for preparation. This would speed up the process of serving diners and ultimately reduce the time taken to occupy a table. Indirectly, the restaurant could improve its profits by serving more diners.

Saving money

Success can also be measured in terms of the amount of money saved. Probably one of the most important reasons for any business to embark on a computing project is to save money and, as a result, to increase profits. Savings can be measured not only in direct costs but also in other ways, such as increased productivity, which allows a business to produce more and increase profits. One obvious example is by replacing human resources with computers, such as robots used in factories in the car industry. Not only does the company save money on labour costs but can also increase the hours worked. However, the risk is the increased cost of maintenance of the robots and the contingency plan if a robot ceases to operate.

Maintaining or increasing profits

As in the restaurant example given earlier, a business might embark on a project to increase their profits or they may need to maintain their current profits if there is a risk of decline or little chance for profit growth. For example, a small, independent retailer might find their profits declining due to new competition from a major supermarket chain.

In order to reduce the decline and maintain their current profits, the owner might embark on a project to put in place systems that will improve their productivity. They may automate the stock control system to provide an instant ordering system with suppliers, which will speed up the time it takes for stock to be replaced so that they can be more competitive with the major supermarket. In this example, the introduction of automation replaces the need to employ additional staff to undertake stock control and ordering, while mitigating wastage either through over-stocking or **expiration**.

Key term

Expiration – in the retail industry, this is the date by which stock must be removed from sale according to the sell-by date.

Improving services

The aim of a project might be to improve services to the customer, which should benefit a business by improving customer satisfaction. The introduction of contactless payment and ApplePay® have sped up the process for customers purchasing low-priced items, such as take-away snacks and drinks, newspapers and some travel tickets.

By speeding up the process, a business can also increase profits because they can serve more customers in the same time and because customers are more likely to come back. Other examples include customer click and collect services such as those provided by major retailers.

Growing the business

The motive for a project might be to grow the business. Growth does not necessarily result in an instant increase in profits but could bring longer term benefits by:

- increasing market share: by extending the way in which customers can purchase products or services, increasing promotion through social media, introducing reward systems and gathering customer feedback.
- improving productivity: by streamlining processes, such as automated packing, customer self-service systems or use of adaptive technology.

Expected return on investment

Every business would expect to make a return on the investment put into a project. An expected return on investment is both justification for the project and a forecast of project success.

Justification for the project

The justification for the project would be a rationale stating the reasons why the project should go ahead and identifying the benefits as a result of the project, including the expected return on investment. The justification is likely to be presented by the project instigator to the key decision makers, such as the business directors and other stakeholders. Businesses distribute the authority to spend against a predetermined budget and, depending on the policies of each business, individuals will know the procedure for progressing a project through to the sign off process.

A forecast of project success

The justification, which is likely to be presented in a report or paper, will need to include a forecast of project success. This forecast will include the expected return on investment and will shape the success criteria that will be used to plan the aims and objectives of the project. It will be used to measure the progress of the project against the milestones and aims of the project. For example a software development project might be measured in terms of financial savings such as reduced costs or increased profit.

The project life cycle

Every project has a life cycle from its inception through to its completion. The project begins with an idea and then, once the project is approved, it cycles through a number of stages, as outlined in Figure 3.3 below.

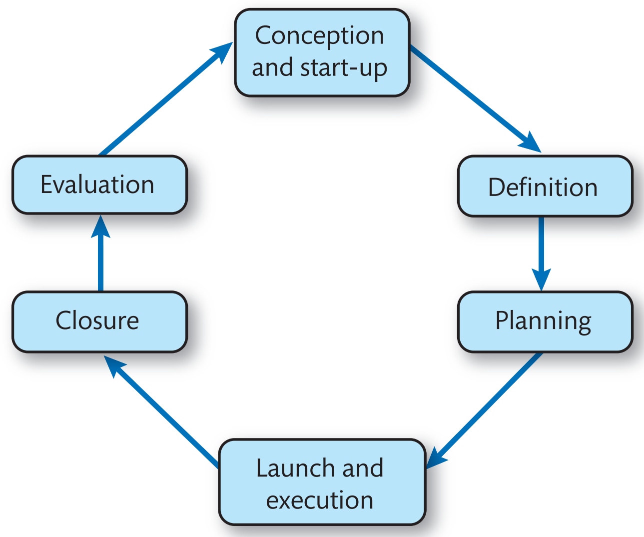


Figure 3.3: Basic stages of the project life cycle

Iterative cycle of project management

Project management can be described as an iterative process. This type of process involves going over the stages of a project again and again to check that they are being carried out as required. For example, Figure 3.3 represents the stages of the project management process from planning through to evaluation, in the form of a one way cycle. This cycle may be repeated on numerous occasions at every stage of the project, for example to adjust or refine the project and its associated resources based on the outcomes of previous stages or benchmarks.

As with any framework, the stages of the iterative cycle of project management provide generic steps that are integrated into a project without prescribing how you actually carry out each stage. Organisations can use or adapt their existing quality procedures rather than having to create something completely new. Quality frameworks also provide guidance to those businesses that need to develop more rigour in their processes.

We will expand on each of these stages in turn. We shall start where an idea is developed, which might begin within

the business, perhaps coming from a member of staff, or outside the business by an idea being sold, in principle, to a business, perhaps to resolve an issue they have already identified.

Conception and start-up

The idea, or concept for a project, might emerge out of a desire to create something or from a complaint, for example a desire to want to reduce the amount of **bureaucracy** in a business or from customers complaining about after sales support.

Key term

Bureaucracy – excessive administrative procedures often generating an overload of paperwork.

Once the idea has emerged, it will need to be conceptualised – in other words, brought to life in the context of the business. The concept will not usually include the detail about how the project will be executed or even what resources will be required to carry out the project. This is the stage where specialists advise what is possible and what is not, or alternative ways to achieve the aim of the project.

The number of people involved will depend on the size of the project and its complexities. Therefore if the project is fairly large and likely to involve many people and carry a costly budget, the project will involve numerous specialists and decision makers. A series of meetings will need to be held in order for the concept to take shape long before the details or schedule are agreed in their final form.

Project mandate

Most projects have some form of project mandate. This is the stage where an idea becomes more than just a concept but before it is fully planned out. The mandate, often in a document of some sort, will respond to questions about the project. These may include:

- the nature of the project – what it is and its purpose
- whether it is feasible – what restrictions exist which may prevent the project from being completed
- financial considerations – whether there is money available to undertake and complete the project
- the proposed timeframe – and whether this presents any difficulties
- the scope of the project – what is necessary and what is only desirable.
- The project mandate provides a sort of starting block so that once these questions have been carefully considered and suitably answered, the starter's orders can be given so that the project can move on to the next stage, assuming the answers confirm its viability and benefit to the business.

Client requirements

To establish the nature of the project, the client needs to be able to express their requirements. This allows the PM to set out the nature of the project. The difficulty here is that, although the client usually knows what they want, in a sense, they are not always specialists in the area of expertise of the project and so the requirements they describe might not be the same as the best solution to their problem.

For example, a client might want a single storey bungalow built to their specifications. The specialist, in this case a structural engineer, might suggest to the client that the building plot is situated in a flood plain and that a house on stilts would be a better proposition. In this case, obviously the client should change their requirements, but in other cases it is more a matter of preference, and the PM and specialists need to give careful consideration to what the client has requested, because they are paying for the project.

It is not unusual to become seduced by a project and, while passion about your work is a positive thing, you will need to be aware that it is easy to unconsciously promote your preferences rather than those of the client. For example, you may have experienced something similar when you have been to the hairdressers and the result was not as you expected – the hairdresser thought they knew what you wanted but really were cutting your hair based on their own preferences.

Project feasibility

The feasibility of the project might decide whether a project is rejected completely or needs to be tackled in a different way in order to achieve the overall aim. Therefore, using the bungalow example, the client will still end up with a single storey building, but it will be raised above the ground to mitigate the risk of flooding.

However, if the client is unable to access the building up steps and the budget will not accommodate a lift or other alternative means to get into the house or cover the cost of putting piles into the ground to support a structure on stilts, the project might not be possible, meaning that it is not feasible.

Definition of the project

Once the project mandate has been agreed in principle and contracts are in place to confirm the agreed terms and conditions, the project can be defined in detail.

Set up the project team

Fundamental to the success of any project is the team assigned to it. The project team is led by the PM and the success of the team will be determined by their skills, knowledge and abilities as well as how they work together as a team. Every team member will need to be very clear about their own role and that of others. They will need to know what their role entails and the impact they have on the project as a whole. In other words, they will need to know where they fit in so that the team, rather like a wheel, has the right amount of cogs in the right places so that it can all work well together.

One of the important considerations when selecting the right project team is the credibility of each individual, for example the testimonials, qualifications and experience they have in the relevant role.

Research

Find out more about Prince2, which is one of the most widely known and highly regarded qualifications for PMs.

Create the Project Initiation Document (PID)

All projects require a PID which defines the project and states its scope. The scope of the project defines the boundaries that the project must work within. The PID should include the following details:

- Purpose and justification of the project.
- Deliverables and success criteria for each one.
- Constraints the project must work within.
- How the project will be approached (carried out) to meet its objectives.
- The structure of the project team.
- The project team's roles descriptions (what each person will do and is responsible for).

Research

Explore the job profiles (roles) of IT project team members. Start by exploring https://nationalcareersservice.direct.gov.uk/advice/planning/jobprofiles/Pages/itprojectmanager.aspx

Consider what project team roles might be required for a project to develop a new computer game.

- Monitoring methods including quality control measures.
- Risk assessments and management.
- Communication methods within the team, including planned project meetings.
- Timeframe, milestones and deadlines.
- Project plan (the schedule for when things will happen).
- Budget broken down into each planned-for cost.

Link

You will learn more about the PID in a computing context in the section 'The Project Initiation Document (PID)'.

Planning

For any project or simple activity, the planning is crucial to the success of the task. Even a social activity requires some planning, even if it is to buy a sandwich. You would decide on the time to purchase, where from, what you want to eat and how much you are prepared to spend. Your plan would include any change of clothing or footwear needed to leave the house, and ensuring that you have your wallet or purse. The plan may change, for example if you decide to shop elsewhere because your preference is not available or if the queue is too long, or you may also decide to buy a drink or something to accompany your sandwich. Afterwards, you are likely to make a judgement about the quality of the sandwich and this evaluation will influence your decision next time you plan to buy a sandwich.

There are numerous software applications that provide templates for project planning and management, such as those offered by Microsoft® and Apple®.

Research

Carry out some research into project planning document templates. Starting by exploring https://www.jisc.ac.uk/full-guide/project-management

Timescales

Most projects are driven by timescales whether out of necessity or simply desire. Once a project idea has been formulated, most clients want to get on with it now rather than wait, unless constraints prevent starting immediately. However, the team carrying out the project may say otherwise due to their ability to schedule the work or the availability of resources, but mainly because the planning stage can take as long as the project itself. Take for example a bespoke database system. The planning stage can take at least 50 per cent of the project time with the actual production taking comparatively little time.

Costs

When planning and executing a project, it is easy to overlook some of the costs that are involved and it is difficult to anticipate everything that might happen, which is why every budget should include a contingency. You may have watched TV programmes about property development where, despite apparently careful planning, the developer almost always goes over their planned budget if they execute the original plan. One problem is trying to manage costs over which you have no influence, such as contractors that you have to bring in. A common oversight is not costing for everything that will be required, for example:

- time, such as that taken up on phone calls and additional journeys
- costs associated with time and for small items of stationery
- administrative services (these are easily overlooked if they are already available but a proportion should be chargeable to a project)
- overheads, such as business premises, equipment, telecommunications, travel costs and wages (all have to be paid for and a proportion should be attributed to a project if they are related in any way).

A project could be agreed for a fixed price or an estimated cost budget. Either way, someone could end up out of pocket if the project is not properly costed or the risks not sufficiently anticipated.

Quality management

Quality management is the process whereby businesses put systems in place to ensure that the quality of their product or service is consistent. You are probably aware of some of the quality systems in place where you study, such as the moderation or verification procedures carried out to confirm that the standard of assessment of your work is reliable and accurate.

Every project should plan and prepare for quality checks which are systematic and frequent. Quality checking will enable the project team to check that the objectives are being met and, if not, make the necessary changes.

Link

For more about quality management, look back at the section 'Quality and deliverables'.

Risk management and controls

You have already started to consider how to anticipating risk and risk management is about dealing with those risks and any that occur unexpectedly. The controls to manage those risks should be clearly set out as procedures before implementation of a project. They may also be guided by government legislation or regulations, such as the codes of conduct set out by industry bodies.

Link

For more about the relevant codes of conduct, see the section 'The codes of conduct developed by professional bodies', in 'Professionalism'.

Link

For more about risk management and controls, look at the section on 'Risk'.

Launch and execution

After thorough planning, the project is eventually launched. This should only happen once all parties have agreed to the PID and the team are prepared and clear about their roles and responsibilities.

You may have been in a situation where you felt unprepared or did not understand what your role was or where you fit in. You may have felt like that when you started your studies and had to learn how to mix with new people, each jostling for their position. If you have ever studied an ant colony, this is a fine example of how a team operates smoothly and efficiently, until an obstacle is placed in its way, when there is chaos! It is extremely important for a PM to make sure that everyone in a project team is clear about their role and responsibilities prior to starting work on a project, because if these things are left unclear problems are likely to arise.

Theory into practice

Produce a plan for a micro (small) project using the steps you have been learning about so far. For example you might plan a party, a gig for a band or a day trip with a group of family or friends. Swap with a peer and critique each other's plans.

Reflect on the difficulties that you or your peer thought would have an impact on the execution of the plan.

Tip

You could also try sharing your personal study action plan with a peer and assess how well they can understand what is expected and how it will be executed.

Carrying out the plan

Providing the planning stage has been thorough and rigorous, carrying out the plan should enable progress to be made towards the end goal of the project. Of course, problems can occur and conditions beyond our control can have an impact on progress, for example the weather and **acts of God**.

Key term

Act of God – an uncontrollable natural occurrence.

Carrying out a plan efficiently and effectively requires the PM and team to state the order that events will occur, while also taking into account the risks of some things not happening in the way they were planned, perhaps due to delays in receiving materials or due to illness in the team. Each member of the team will need to work from the plan and report any deviations to the PM so that they can be accounted for and incorporated into the updated plan.

Monitoring activity

Every plan requires monitoring to review its effectiveness and progress towards meeting objectives within the timeframe. Just as your progress in our studies is monitored by your teacher and by yourself to ensure that you meet your deadlines, PMs will need to monitor the effectiveness of teams, evaluate what is working well and where gaps or weaknesses appear. Plans may need to be changed and milestones shifted.

Reflect

How often do you monitor your own progress against your study plan to ensure that you meet your deadlines?

Checking progress

Project teams hold frequent meetings to check on the progress against the planned expectations and milestones. Every meeting takes time and costs money and unexpected meetings may impact on the budget. You should also consider whether it is effective to have more meetings, as they take the team away from actually working on the project tasks.

To check progress, the plan will identify every stage of the project and include a criterion to measure whether the objective is on schedule. Budgets will be checked against the progress made and plans will be adjusted accordingly. It is rare to find projects where the work was completed on time and within budget, especially very large ones such as the building of the Olympic Stadium for the 2012 London Olympics.

Reflect

What techniques do you and your teacher use to check on your progress? What could you do to improve those techniques so that you can work more efficiently?

Tip

Try breaking down your targets into smaller steps with deadlines. If your milestone dates are more than a week apart, it probably means that your action is too large to manage in one go and needs to be broken into smaller steps.

Closure

Once all the stages of the project have been executed according to the PID, the client and PM prepare for sign off and closure. For the stages of the project to have been executed, all of the deliverables of the project must have been created or carried out successfully.

Handover of the product

The handover of the project will occur after all relevant stakeholders have agreed that the deliverables have been completed to their satisfaction. This will rely on a number of quality management assessments.

User acceptance testing

Periods of testing are likely to take place during the implementation of the project (product/service/system). Users will be involved in the testing to establish whether or not the product meets their requirements. The purpose of user acceptance testing is to establish whether the product meets the user's requirements within the limitations of the client's requirements for the deliverables, although these should have been based on user requirements in the first place if the product is to be of benefit to users and therefore successful.

PMs should establish who should be involved in the user acceptance testing and what stages are being tested. (As well as testing the final product, users can be asked to test the concept, plan/design and early versions of a product.) In many cases, users are not likely to be asked to test the whole system but rather just a **module** of it. During development, each module is likely to be built and tested independently, often by different groups of developers and testers.

Key term

Module – A part of a large software system that carries out a specific function. For example, different departments will use different modules within a full IT system: the Human Resources department will use a payroll module to calculate staff wages.

Once user acceptance testing has been carried out on the fully implemented system or completed product/service, any resulting feedback should be considered and incorporated into the system, if possible. Then, once everyone is agreed that the deliverables have been completed, the project can be closed.

Disbanding project team

When a project ends, the team might be going their separate ways, especially if they are sub-contracted and not employed by the client business or PM. As we learned earlier, the team should be put together based on what each individual can provide to the successful execution of the project and not because they are available at the right time or place.

There are approaches for disbanding the project team which include arrangements for:

- thanking the team
- giving feedback based on an evaluation of their performance as individuals and as a team
- handover by the team of anything outstanding, including documentation, certification for the client, manuals, equipment and knowledge
- future contact should the need arise during any guarantee period (these arrangements should be built into the contracts of team members before embarking on the project)
- availability and capacity for possible future projects
- transition to the next project if they will be working with the same PM again.

Post-project evaluation

As with any activity, regardless of size, we can learn from evaluating its success and use those lessons for future projects. Whether it is general human nature or dependent on personality, we tend to focus on the faults and overlook the positives. You possibly do that when reviewing your own assignments or interim projects or even your own behaviour.

However, if we overlook what went well and do not analyse why it went well, we may lose valuable experience. If you do not scrutinise the reasons why and how something went well then how will you know to replicate it in the future? The great advantage of learning by experience is that you can apply the positive experiences to skills in different contexts, and on different projects. These are known as transferable skills.

Tip

There are several tools or templates that you might find useful to help you to identify what went well and what needs improving on future projects. One simple method is a SWOT analysis.

Research

Research SWOT analysis and consider how you might apply it to your post-project evaluations.

Reviewing the project against success criteria

When you evaluate the success of a project you should do so against the success criteria that you set out at the start of the project, just as you would when writing an assignment, peer assessing and reviewing your own achievements. Measuring against the success criteria enables the PM and the project team to evaluate how improvements could be made on future projects and how what worked well can be replicated in other settings. One way of representing a framework for evaluation could be as a Venn diagram where the success criteria are central to all aspects of the project review, as shown in Figure 3.4.

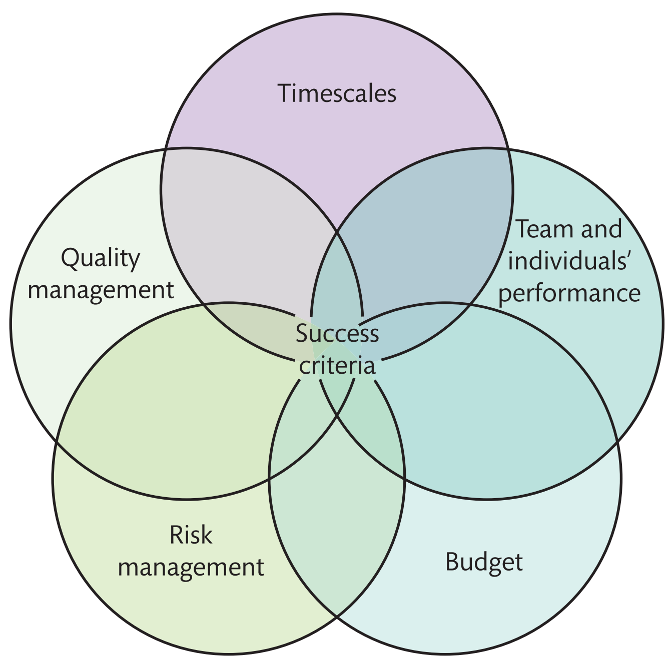


Figure 3.4: Post-project evaluation represented as a diagram

Through each project and its review, the PM and project team can develop a wealth of knowledge from their experiences which will, for example, enable them to assess the risks more accurately in the future and to consider the previously unconsidered. The saying 'hindsight is a wonderful thing' can be a useful one when taking those **lessons learned** and applying them to a variety of contexts in the future.

Key term

Lessons learned – insights gained from an experience which can be usefully applied to future projects.

Discussion

In a small group, consider one example of a time when you did not learn from previous experience and discuss what happened, and then think of another example where you took a lesson learned from one experience and applied it in a later situation.

Professionalism

The credibility of the project team was considered in a previous section and, although qualifications do confirm an achievement, the integrity of an individual is enhanced by their professionalism and how they apply their knowledge and understanding to their work.

Professionalism is demonstrated in many ways. As you begin to learn more about codes of conduct regulated by relevant professional bodies in this section, you might also find it useful to study customer service skills to learn more about the characteristics that are crucial to operating as a successful professional in the computing industry.

Research

Carry out research into the personal and interpersonal characteristics that are vital to good customer service.

Discussion

In a small group or with a peer, discuss what professionalism means to you and how it can be verified.

The codes of conduct developed by professional bodies

This section will look at the codes of conduct developed by professional bodies and their impact on how a project is planned and managed in an ethical way. Professional bodies enforce legislation relating to their sector of expertise by prescribing a code of conduct or practice. Membership of these bodies is beneficial to members as it is recognised widely by the industry and provides a degree of integrity and standardisation.

What is meant by planning and managing a project in an ethical way is complex and viewed by many professional bodies as a core skill. Some of the characteristics of behaving ethically include:

- truthfulness and honesty
- integrity
- respectfulness
- being environmentally considerate
- punctuality, i.e. being on time.

Link

For more about ethics in project management view the articles at https://www.apm.org.uk/search/site/ethical

Three professional bodies that have codes of conduct relevant to project management in the computing industry are discussed below.

Association of Project Management (APM)

The APM is the largest professional body for project management in Europe and it provides services to members relating to project management and development. APM provides resources to its members such as current legislation and updates that impact on project management services relating to areas such as risk, finance and quality management.

Link

To learn more about the APM, visit their website https://www.apm.org.uk/

Amongst the guidance and resources provided by APM is their FIVE Dimensions of Professionalism model.

Each of the five dimensions (sections) measures professionalism by the individual or teams. These are:

- breadth (of knowledge)
- depth (how knowledge and understanding is applied as competency)
- achievement (qualifications to confirm competency)
- commitment (to your continuous development, practice and career progression)
- accountability (to practising ethically) – referred to as the 'Code of Professional Conduct'.

British Computer Society (BCS)

The BCS, like APM, recognises a range of qualifications in support of the role of PM or team member (one of which is PRINCE2, which was introduced to you earlier). The BCS provides training and certification to regulate the quality of IT and computing services, which is recognised in more than 200 countries. Employers and contractors recognise the training and certification of the BCS as a demonstration of skills and validation that recognised industry standards are being implemented by that individual or team.

Link

For more about the BCS, visit their website http://www.bcs.org/

Project Management Institute (PMI)

The third relevant professional body is the PMI which, like the other two bodies, provides training, certification and membership of a professional body that is recognised globally for setting and maintaining standards. Just like the other two professional bodies, PMI provides a benchmark for what is deemed acceptable conduct and how to maintain and exceed the expectations for project quality and professionalism in an ever-changing technological world.

Link

For more about the PMI, visit their website http://www.pmi.org/

Communication and presentation for project planning and management activities

Effective communication is often said to be the key to success. In this section, we shall begin to explore what is meant by effective communication and its impact on project planning and management activities associated with the project. We shall also consider how the presentation of information impacts on project planning and the management activities of a PM.

Appropriate for target audience

You may have heard the phrase 'being appropriate for the target audience' and this can apply to communication. What this means is that whoever is involved, in whatever role, whether a client or a team member, the methods of communication can make or break the project. In Table 3.5, the stages of the project life cycle are considered and appropriate methods of communication are outlined.

However, even Table 3.5 should only be considered guidance because the best methods for communicating may come down to the individuals in question. How often each method of communication is needed at each stage of the project should also be considered. For example, the frequency (regularity) of face-to-face meetings needed at each stage of the project will vary.

Conveys intended meaning

One communication difficulty frequently encountered is how to articulate a point in such a way that the person on the receiving end (the target audience) interprets the message accurately.

Table 3.5: Methods for communicating with different target audiences relating to the project life cycle stages (and methods of communication legend)

Target audience	Conception and start-up	Definition of project	Planning	Launch and execution	Closure	Post-project evaluation
Project team	RI	F2F, EM, PC	F2F, EM, PC, FD	F2F, FD, PC, EM	F2F, FD	F2F
PM	RI, F2F	F2F, EM, PC, FD	F2F, EM, PC, FD	F2F, FD, PC, EM	F2F, FD	F2F, FD
Client	RI, F2F	F2F, EM, PC, FD	F2F, FD	F2F, FD, PC, EM	F2F, FD	F2F, FD

Methods of communication legend	
RI	Rough ideas
F2F	Face-to-face meetings
FD	Formal document
EM	Email
PC	Phone conversation

You may well have experienced this when you have been shopping for a new gadget or even searching on the internet and the response to your search request generates a response unlike your request. We all have a different view of the world and, to add to the difficulties, we do not all speak the same language and we do not understand technical terms to the same extent or in the same way.

Theory into practice

Identify a technical problem you have encountered and seek help from technical support. Try using a variety of methods of communication to resolve the issue. Perhaps try an internet site, the helpline and support from a supplier. Compare the outcomes to determine which methods were most successful and why. You will find it useful to make notes. Discuss your experiences with a peer.

Think about it

- What will you do differently another time?
- How would you handle the situation if you were the person on the support side?

Effective use of graphics to support meaning

It is widely believed that we each have our own preferred way of learning. This does not mean that we should only ever use that way of learning as we would never extend or stretch ourselves and this would limit the kinds of new skills we could learn. However, it is important to recognise that, if we are presenting information to other people, we should be mindful of their preferences.

In business, we want to make sure that our target audience fully understands what we are trying to say and, in the case of a project, we want to ensure that there is no misunderstanding which may result in the client getting something different from their expectations in terms of the deliverables, cost or deadline.

Therefore it can be useful to present information visually as well as textually. Text or audio narrative should support the graphics to expand on the meaning because graphics alone can be misleading.

Instructions for surfing the internet:

Open a web browser and type the URL of the website you want to visit in the navigation bar.

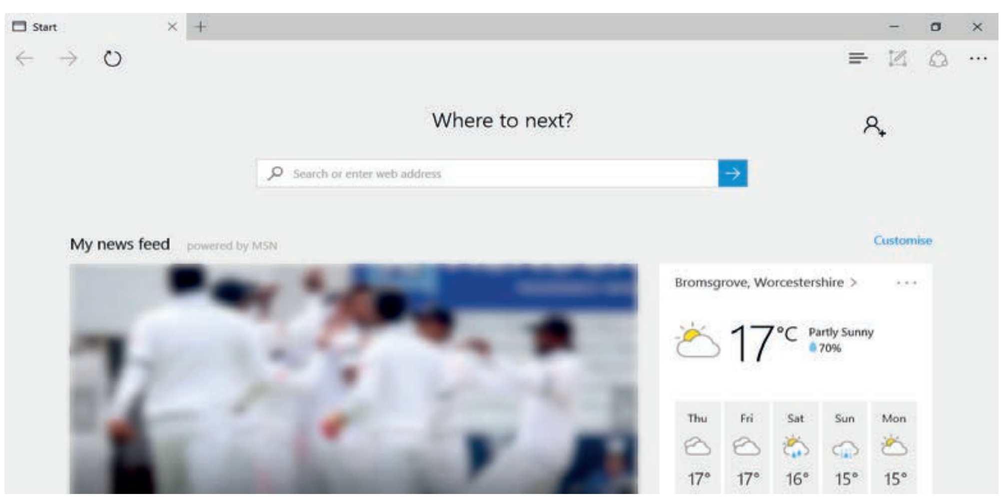


Figure 3.5: What's wrong with the instruction provided for explaining how to surf the internet? How useful is the image without any instructions?

You will be aware that graphics are often used to catch your attention but they also help to break up lengthy text. You need to be mindful that some people are unable to read text easily, especially if they have certain disabilities such as dyslexia; just as images will be hard to interpret for people with impaired vision including colour blindness.

Use of fluent English and appropriate technical language

Fluency refers to your knowledge and use of a wide range of vocabulary, which enables you to use alternative words to suit different audiences. For example, if a meteorologist wanted to tell a four-year-old child about the extreme weather conditions, they would probably use different words than if they were explaining the same conditions to an adult. They would simplify the language so that the child could understand more easily.

As mentioned earlier, when you start introducing technical language, whether spoken or written, the chances of being misunderstood or not understood at all are increased considerably. When working on any project, it is important to ask the right questions to draw out information that you otherwise would not know. Unfortunately clients will not automatically know what a PM wants to know, and their interaction becomes even more complex when they are faced with language barriers or cultural differences such as accents and **colloquialisms**.

Appropriate tone for project documentation

If you have ever received a text message or email written in capital letters you will have begun to appreciate that tone does not only exist in **audible** messages. It is easy to overlook the way in which a message can convey different meanings depending on the way it is written (for example in capitals) and the words that we use. This is why it is so important that you proofread everything you write and why it is helpful to get someone else to read what you produce, as you are likely to read what you expect to read rather than what is actually written.

Unfortunately spellcheckers on word-processed documents are often misused and it is assumed that they can replace the need to proofread. Although a computer spellchecker is useful, it does not read, and a spelling or grammar check does not necessarily accurately reflect what has always been referred to as **Queen's English**. An added complication is that any word, even if misspelt, can be added to the dictionary even without the knowledge of the person creating the document.

Clear communication is essential for professional interaction with other people.

Key terms

Colloquialisms – words and phrases that are local to the region where you live and which are not grammatically accurate, such as 'I should of...' instead of 'I should have...'.

Audible – a sound such as speech, music or noise.

Queen's English – spoken and written English language, the standard agreed for formal communication in Britain.

PAUSE POINT How fluent is your English language?

Hint Think about any colloquialisms you might use and how clearly you articulate words and phrases.

Extend Compare how you speak with how you would write the same phrase. Rewrite the same phrase to suit different audiences such as a friend, parent, teacher and employer. Would you change anything if you were trying to impress someone, perhaps at an interview?

Case study

T-Systems

T-Systems operates information and communication technology (ICT) systems for multinational corporations and public sector institutions. It has offices in more than 20 countries and approximately 50,000 employees who have expertise in ICT innovations and it generates an annual revenue of Euros 9.5 billion.

Travel operator TUI contracted with T-Systems to manage their IT project to consolidate their systems. Their objectives were to increase collaboration between employees and TUI's European markets, improve the customer experience and rationalise the supplier base while delivering significant cost savings to TUI. The transformation would be a significant business change programme that would involve the contact centres, retail high street stores, offices and also airport locations.

This was a complex project and, as each business had different requirements, it meant that decision making was difficult. Some parts of the business were less involved in the project discussions than others and that resulted in ideas being rejected when the criteria for the project could not be agreed upon by everyone. Following extensive, collaborative discussions, there were significant changes to the project plan, including the countries included in the scope of the deal, the commercial model and the contractual terms for the programme.

The execution and deployment was led by an overall transformation programme, with senior project managers responsible for key work streams that aligned with the programme, test managers and roll-out managers. Stakeholder management remained critical to success. Setting up working groups, that included T-Systems, TUI IT, TUI business people and led by the TUI change team, proved invaluable and increased the vital flow of communication between all stakeholders in contact centres, shops, offices and airports.

The project was back on track and the schedule revised, which resulted in TUI being the first travel operator to implement a single form of communication between all parties. Customer satisfaction scores were used to measure the success of the project, which was recorded as globally significant by the end of the first year following implementation.

Check your knowledge

- What was the purpose of this project?
- What role did the project team have in the success of this project?
- Who can you identify as stakeholders in this project?
- What risks can you identify?
- What did the project team do to resolve issues?
- What are the benefits resulting from this project?

PAUSE POINT What was the life cycle of the TUI project?

Hint Try drawing the stages of the project to represent the project life cycle.

Extend If you were the PM for the project in the case study, how would you go about planning the project?

Theory into practice

Identify a project where a business has implemented a new IT system. Read the case study and identify each stage of the project by comparing with the project life cycle.

Tip

Keep all your working and notes for this case study and, after working through section B, revisit it to identify where you could develop the project further.

You could try again after completing the other sections too. You may have to make some assumptions where detail is lacking.

Assessment practice 3.1

A01 A02

You have a part-time job in a bakery and, although they are a local business, they have many recipes that are revered by the customers as being 'second to none'. As the recipes have been handed down from generation to generation, the newest member of the family to take over the family concern, Jason, has decided, as he has no plans to have a family, that the recipes might become lost as there may not be a future generation to hand them down to.

Jason is forward thinking and plans to have all the recipes captured on an IT system, together with images and instructions of how to make each product and where the ingredients are sourced. Up to now, his father has run the business according to his traditional, old fashioned values and processes, keeping everything in his head and managing business transactions over the telephone and with minimal paperwork.

Jason wants to ensure that the family recipes do not get lost should anything happen to him or his father. Jason is quite open to new ideas and will even consider selling the recipes or perhaps writing some books when he has the time or recognises a lucrative gap in the market.

He has asked you to help him by forming a project team to instigate this idea. This is what you need to do as his project manager.

1. Conceptualise the life cycle of his project idea.
2. Identify the risks associated with embarking on such a project.
3. Outline a justification for the project for Jason to present to his father who, as the major shareholder of the business, will be the main person involved in any decision making.
4. Prepare a simplified presentation that both he and his father can understand, which clearly demonstrates the life cycle of the project and any obstacles to be overcome.

Plan
- What are the main facts I have been given?
- Where will I start?
- What do I already know and what do I need to understand better?
- What further sources of information will be useful to me?

Do
- I know how to locate where my weaker areas are and how to improve.
- I understand what is being asked of me.
- I can check that the criteria are all met.
- I can describe improvements to my analytical and evaluating skills.

Review
- I can describe how to improve further.
- I can recognise my strengths.
- I can explain how I tackled each part of this activity.
- I can explain how this knowledge and experience will be useful to me elsewhere.

PAUSE POINT What did you enjoy most about this task? What did you find the hardest?

Hint Keep notes of your thoughts along with the dates, so that you can look back at a later date and measure how much progress you have made.

Extend Provide Jason with some suggestions about how the project will be quality assured and suggestions for contingencies to mitigate the risks already identified.

B Starting up a computing project

When any form of project is considered, there are steps that should be undertaken to ensure that the idea for the project is feasible. This involves gathering the key information needed to run a successful project, the production of the PID and obtaining authorisation for the **project kick-off**.

Key term

Project kick-off – the official launch of the project and the point at which details of the project are promoted.

The project kick-off will only happen after some initial investigation to establish that the project is viable. This involves asking the following questions:

- Can the client afford it?
- Can it be done within the timescale?
- Is it technically possible?

In most cases, although the project may be technically feasible, the cost and the timescale are often greatly underestimated. For example the building of High Speed 2 (HS2) is a high-profile long-term project which, when underway, might well cost considerably more than the original estimate of £32 billion (having already increased by around 30 per cent) and take longer than suggested.

Interpreting the business case

The **business case** is the driver of the project. An idea may have been formed on the basis of a smarter way of working which aims to improve the efficiency of business operations – this is the reason for carrying out a project, i.e. the business case for doing so.

Key term

Business case – justification of a project according to the benefits it will bring to business functions and profits.

For example, one such business case was the introduction of self-service checkouts in UK supermarkets in 2011 whereby checkout assistants are no longer needed at every till because shoppers can scan their own items rather than having to wait for the assistant to do so for them. Retailers promoted the case for self-service checkouts by telling shoppers that they could scan their shopping at one of several self-service points rather than wait in queues for service from assistants. Some may say that this makes paying for shopping more efficient, partly because they do not need to engage in conversation with the employee, or because the shopper is more involved in the transaction. However some find the experience difficult to understand and still require human assistance so it is not always a more efficient experience for them.

Reasons for the project

An idea for a project needs to be supported by good reasons for it to be implemented and a business case needs to be drawn up. The reason for the Government wanting to go ahead with the high speed railway line project HS2, for example, is to 'improve links between the north and south of England...[as] the best way to meet the challenge of an increasingly overcrowded transport network' and the project is promoted as 'an investment for the future'. This is what the UK Government refers to as the 'strategic case for HS2'. In other words, this is their business case which sets out their aims and how they intend to achieve a long-term vision.

Link

To find out more about the Government's 'strategic case for HS2', visit: https://www.gov.uk/government/collections/the-strategic-case-for-hs2

Even on a project of a much smaller scale, there has to be a reason for it. An example might be implementing a spreadsheet for managing a household budget or designing a database to contain the contact details and birthdays of your friends and relatives. The reason for creating them is that they will give you some benefit, either to help you manage your finances better or to be more organised when it comes to sending birthday cards.

Options that should be considered

As you started to explore earlier in this section, there are multiple factors to consider when planning a project. Planning is possibly the most important stage of any project. One of the main pitfalls of projects can be the eagerness to throw yourself straight into the project before fully planning it out.

You could probably imagine how disastrous that would be if you were, for example, lucky enough to build your own house or have one built for you. You may even have seen some of the television programmes where people aim to build their dream home without using a PM or even employing an architect to produce detailed drawings and a specification. While their efforts might result in a building, it will come at a considerable price often exceeding their original budget by a significant amount and will involve compromises on some of the finishes or features they originally desired.

With a computing project, just as with building a house or launching any other project idea, if the planning is not done thoroughly, the result is likely to be disappointing in some areas. For example, if you are creating a database of your family and friends, it is unlikely to perform the range of functions you want it to unless you plan it carefully first. As you plan a project you will also need to consider alternative solutions to the problem that the project addresses. For example you might consider developing a database by modifying an existing application or you might look at developing it from scratch.

When planning a computing project it is important that you consider the following:

- Project goals – what you are you aiming to create.
- Project deliverables – a list of things that need to be created in order to achieve the goal.
- Project schedule – the timeframe for completion, so that the stages of the project can be mapped against the deadline and the number of days/hours to achieve each stage and the resources allocated to each task (team members, materials and equipment) can be calculated.
- Project plan – which includes:
  - details of the resources required and the costs allocated to each one
  - a plan for communicating with all those involved in the project and a schedule of times (such as dates of meetings) for reporting back on progress against milestones
  - a risk management plan which recognises the risks involved and the strategies for mitigating those risks.

While this list might seem a little daunting, the format is one which you can (and may already) use when tackling an assignment or embarking on exam revision. Consider it as a formula which, when followed rigorously, is more likely to result in a successful outcome.

Reflect

Do you use a similar format for planning when you embark on any form of project, no matter how small or large? Do you use anything to help you **formalise** it such as a planning schedule or chart?

Key term

Formalise – turning an informal (relaxed) arrangement into a formal one (by following convention) perhaps by using documents or templates and charts. Gantt charts are often used in project planning.

Link

For more about Gantt charts, see the section, 'Gantt charts as a planning and progress tracking tool', in 'Scheduling and milestones'.

Expected business benefits

To justify a project, you will need to identify the benefits to the business, just as you could probably identify the benefits to yourself of producing an automated spreadsheet for birthdays of family and friends. Businesses are constantly monitoring their profits and seeking ways to increase their profits by becoming more efficient. For example, Kameoka Plant, a Japanese farm near Kyoto, is run entirely without farmers, by using robots to carry out all of the work. They estimate that the benefit to their business of automating their processes will be an increase in productivity from an average of 30,000 heads of lettuce to 10 million heads a year. Since the introduction of **CAD** and **CAM** IT systems, car designers and manufacturers have also increased their efficiency.

Link

To learn more about the Kameoka Plant, visit http://spread.co.jp/en/factory/

Key terms

CAD – computer-aided design software which is used for detailed technical drawings. It is used to help design cars and buildings.

CAM – computer-aided manufacturing is done by machines which use software to operate tools and machinery for manufacturing, so providing greater precision and accuracy.

Tip

If you have not already got a mentor, try to find someone from a local business. Perhaps your place of study can help you find a mentor by contacting some of the businesses they work closely with for apprenticeships or work experience.

Timescale, including major milestones

A little earlier in this section we referred to milestones as a measure of the project progress. A milestone is a measurable point in the development of a project which is planned in advance so that progress towards the deliverables can be measured accurately. You may have milestones which you have discussed and agreed with your teacher – these might be referred to as short- or medium-term targets in your action plans or individual learning plans.

Setting milestones and forward planning is part of putting a schedule together, which assists in systematically preparing everything you need to consider when project planning. The aim of creating a schedule is to enable your project to run more smoothly. The type of schedule you use in your studies involves constructing an action plan.

However, one of the failings of any form of action plan is where targets are set but are not associated with any form of outcome to measure them against. Therefore what happens is that a measurement is associated retrospectively or, alternatively, the outcome is decided upon but there is no specific timescale identified and the targets are often labelled as 'ongoing'.

All action plans need to include **SMART targets** and each aspect, whether it is the time or the resource needed, should be SMART. If a plan just states a time as being ongoing, this does not meet the criteria of being SMART. The time needs to be specific, measurable, achievable and realistic otherwise progress cannot be measured. The milestones of a project should be SMART targets.

Key term

SMART target – a specific, measurable, achievable, realistic and timebound objective of a project.

Link

For more about SMART targets, look back at the section 'Defining success criteria and using SMART objectives to define project outcomes'.

Budget available

Every project has costs associated with it. Before a project is planned, the budget must be calculated. This will be based on as much information as possible about the resources (team members, materials and equipment) needed to complete the project. As you plan a project, you will work back from the budget to allocate funds for each activity and associated resources. A budget must be set in advance otherwise the costs can spiral out of control. When allocating the costs to every aspect of the project, it becomes clearer why projects often cost more than anticipated. It is at this stage, if not before, that discussions with the client will entail ways to reduce the costs, perhaps by clarifying the 'must-haves' of a project and those which are simply on the wish list.

Project leaders are likely to want to set aside a sum of money as a contingency for costs which might be unexpected. These could arise because of increases in the cost of materials or wages or because the project has gone over the expected time.

Hidden costs such as time are often overlooked. Time is not free, as the time spent by each team member needs to be paid for in terms of their wages and the equipment (computers and software) that they need to fulfil their roles on the project.

Another example of hidden costs is transport and travel. It is easy to overlook the running costs involved. Every trip costs money and rising fuel costs and making an unnecessary number of journeys can add to the costs of a project.

Major risks

There are major risks associated with any project and the severity of each risk will need to be determined to assess whether or not it will have a major impact. Of course, a risk is major if it is life threatening.

Therefore, for example, an IT project for the NHS which interrupts the schedule of operations or interferes with the systems used to diagnose health concerns will have major risks attached to it. In contrast, the risk associated with your friends and family database is likely to be the potential loss of data and this can easily be mitigated by keeping a backup copy. The hospital IT project would require considerable planning, thought and discussion to negotiate ways to mitigate the risk to patients, perhaps by running the existing processes and systems alongside the new system until it has been thoroughly tested for reliability.

Tour operator Thomas Cook experienced some disruptions with accessing customer airline bookings during the installation of their new IT system in late 2015 and early 2016. According to their automated telephone message, bookings made prior to November 2015 were temporarily inaccessible to staff although passengers could still make changes via the Thomas Cook website

Link

For more information about the issues Thomas Cook experienced, visit http://www.isitdownrightnow.com/thomascook.com.html

Tip

You can check out the performance of any website at http://www.isitdownrightnow.com

Stakeholders

Stakeholders are anyone with a vested interest in a project. Project leaders would need to identify anyone involved in or affected by the project and, in the planning stage, allocate their project responsibilities. You could say that we are all stakeholders in the HS2 project as it is a public service funded ultimately by the public purse which means taxpayers and users. This is a more complicated example as one of our responsibilities in the HS2 project is to vote

for or against and give our reasons. On the other hand those stakeholders involved in actually bringing the project to fruition, the construction workers themselves, would have different responsibilities. Either way we are all time constrained in some way or another, and affected by the budget even if only indirectly.

A smaller computing project, such as the introduction of an automated invoicing and payroll system for a small firm of accountants would include in its list of stakeholders:

- the accountants themselves
- any trustees of the business
- any of the employees
- and, most importantly, their clients.

Less obvious stakeholders are their suppliers, the landlord or owner of their premises, the bank where income is managed and, ultimately, Her Majesty's Revenue and Customs (HMRC). The project team are also stakeholders. If the project interrupts the process of the accountancy firm paying their employees, generating invoices for clients and lodging tax returns or paying taxes, then there are many stakeholders who would be very unhappy. This would cost the firm money in lost or delayed income and fines for late payment of taxes, rent or mortgage on the premises. It would also have a negative impact on their reputation.

Key stakeholder responsibilities

Each key stakeholder has a responsibility for their contribution to the successful execution of the project. One particularly important factor in the failure of projects, no matter how small, is when the team members do not understand their role sufficiently or the responsibilities associated with it. Anyone unwilling to take responsibility for their actions is not an asset to the team and is unlikely to be welcome unless they can prove otherwise.

Reflect

Have you ever been part of a team where not everyone has pulled their weight or appeared to shirk their responsibilities? Why do you think this was the case and what could you do to mitigate this occurring in the future?

Project manager – responsible for defining, planning, controlling and leadership

The PM has overall responsibility for the entire project and for the team working on it. They have the authority to make decisions although, democratically, these decisions are likely to involve the client and the team unless time is of the essence or the decision does not warrant involving other stakeholders unnecessarily.

The PM is responsible for defining the project in detail and taking it through the planning stage. They will obviously involve others in the process. Defining the project will involve:

- the client – who will outline the business requirements for the project, its aim(s) and the required deliverables
- the team – who will be able to advise on how the requirements can be fulfilled and the tasks that will need to be carried out to complete the project.

The PM's role is to:

- define and plan the project
- control all aspects of the project, including managing how the budget is spent, the performance of team members and meeting milestones in the schedule
- providing a leadership role by directing, managing and controlling the overall operation of the project.

For example, if you were planning a system for a friend (the client) who wanted to automate their friends and family contact details to generate labels for sending cards for special occasions, you would liaise with them about the different software options to establish their familiarity with and ability to operate the system.

Technical teams – responsible for performing the project tasks

Whereas the PM has overall responsibility for the successful execution of the entire project, the team, in this case referred to as the technical team, is responsible for performing the project tasks.

The PM will clearly define each team member's role in accordance with their technical strengths and skills. Usually, it is up to the team members themselves to allocate the tasks that each will take on, as they are the ones with the expertise to know who is best equipped to carry out each task at any given time.

The team may be employed by the company who is initiating the project or they might be employed or sub-contracted by the PM.

Theory into practice

Next time you are part of a team, perhaps carrying out an activity set by your teacher, dedicate time before beginning it to determine what roles you will each take on, by considering your strengths and abilities. You are likely to discover that this is time well spent and much less stressful or chaotic than the alternative. (This is not to suggest that you should not take on roles that will help you to improve your skills, but in a project team your development can be planned whereas, in a one-off discrete activity, it is often best to play to your strengths for the benefit of the whole team.)

Team managers – responsible for following company policies and providing resources

Depending on the size of the project team, there is likely to be at least one team manager, just as managers exist in companies and in the place where you study. They have an operational, or performing, role but can also provide a conduit (link) between those in the technical team and the PM.

Team managers are responsible for following company policies and providing resources. They monitor the work of the technical team to ensure that it is being carried out in accordance with expectations and the project plan. You can relate this to your teacher monitoring your progress and keeping you on track, or perhaps you have a part-time job where you have a line manager or supervisor.

Project sponsor – provides the authority and guidance, and maintains the priority of the project in the organisation

The project sponsor is the person who owns the project, and they may or may not be employed by the organisation where the project is being implemented. For example, an architect would have the power to go ahead and initiate a project for a new housing development but they are probably acting on behalf of the land owner or construction company. Where the project sponsor is employed by the company, they are likely to be a director or someone in authority.

The project sponsor will provide the guidance as well as maintain authority over the project. They will be responsible for ensuring that the project remains a priority within the business, especially where the project team or PM have several other projects they are working on simultaneously. The project sponsor is someone that the PM can defer to with respect to escalating major risks, which, for example, might require signing off an increase to the budget.

Client – provides the product requirements and project finance

The client is the one who provides the PM with the product requirements and finance for the project. The client might be a business, who will be represented on the project by a project sponsor, or an individual who has hired the PM. The requirements supplied by the client will not include the detail of how the project will be designed and often may not involve the technical knowledge or expertise to explain what is needed.

The client is the one who will have an idea about what they want and, although that is not necessarily the same as what is needed, the project plan can be devised between the client and PM, with input from the team who have the technical expertise. Interpretation of the client's requirements is the responsibility of the technical team and the PM.

Other stakeholders

Previously, you were introduced to the idea that stakeholders are not just those involved in the project but also those affected by the project. Here are some examples of who those other stakeholders might be.

Suppliers – provide materials and equipment

These are the businesses that provide materials and equipment for the project. For example, in the case study about TUI Travel, the contractors would be the software developers and the suppliers would include manufacturers of hardware and suppliers of raw materials to produce the hardware, cabling and other resources identified in the project plan.

Contractors – contribute specialist work

The contractors are those who contribute to the specialist work. They may be members of the project team who are contracted from outside the business. In a computing project, they are likely to have technical IT skills but, in other projects, they might be builders who carry out construction work or even on-site caterers. When major projects are undertaken, such as major building developments, contractors would provide and manage on-site facilities such as portable toilets and skips for the rubbish.

General public – may be affected by the project

The general public may be affected by the project, as in the case of the HS2 project. Thousands of people across the intended HS2 route are likely to be affected, because of the loss in value of their homes, the disruption to the surrounding countryside and threats to wildlife or because they are forced to move out of their homes by way of a compulsory purchase order imposed by the Government to release the land on which their homes are built. Even those who do not live within the line of HS2 are affected by the construction work which will be undertaken, as it may make their journeys to work or taking children to school more difficult, increase atmospheric pollution and create disruption to some public services or utilities.

You may have seen occasions where members of the public form protest groups to express their views more vehemently with regards to projects like HS2. Sometimes these protests can result in projects being halted and even cancelled. Some people see it as their responsibility to protect their homes and surrounding environment in this way.

Conversely, of course HS2 should, in the long term, have a positive benefit on members of the general public who will benefit through using the new faster train line. However, these are not necessarily the same people who are adversely affected by the project.

Identifying assumptions and constraints

It may be necessary to make **assumptions** about certain aspects of a project. Making assumptions might appear to be somewhat reckless but, if carefully thought through, can be very useful. Making assumptions can enable the PM and technical team to identify the constraints (limitations) prior to carrying out the project.

Key term

Assumption – something that is accepted as true or as certain to happen, without evidence. However, reasonable assumptions can be made based on prior experience and knowledge.

Dealing with assumptions as low-level risks documented at the outset

Assumptions can be effectively used in planning a project by dealing with them as low-level risks which are documented at the outset. These low-level risks can be mitigated through contingency plans. By making such assumptions, as long as they are reasonable, more efficient progress can be made on a project.

For example, in Assessment practice 3.1, assumptions would need to be made about Jason's idea to possibly sell his family's recipes in the future. To ensure that the project outcome has longevity, it may be necessary to assume that Jason will want to sell the recipes, so the risks associated with this can be identified and mitigated from the outset.

Case study

What a tragic disaster

The launch of the NASA Space Shuttle Challenger in 1986 ended tragically as a result of inaccurate assumptions made during the project.

On the morning of the launch, crowds gathered and tension mounted as the astronauts prepared to embark on the space mission of their lives. Seven American astronauts, five male and two female, plus one civilian, the high school teacher Christa McAuliffe, had undergone rigorous training in preparation for their flight to orbit the Earth in the Space Shuttle Challenger. It was an unusually cold January morning for the state of Florida, where NASA has its space centre at Cape Canaveral. The planned launch date had already been delayed by six days due to bad weather and technical problems.

Pressure from various stakeholders about the costs of the project due to delays overrode the warnings given by engineers about the extremely low temperatures and the effect they might have on the rubber O-rings, which sealed the rocket boosters. The warnings were not heeded, and this had catastrophic consequences because the seven Americans were killed within seconds of take-off. An incorrect assumption had been made that the outside temperature would not drop that low in Florida.

Check your knowledge

- Explain the reasons for this disaster.
- What assumptions were made that prevented the risk being taken seriously?
- What actions could have been implemented to mitigate the risks of these assumptions?

Research

For more information about the Challenger disaster, visit http://er.jsc.nasa.gov/seh/explode.html

You can read more, listen to and watch videos about the Challenger disaster at http://www.history.com/topics/challenger-disaster.

Reflect

What measures would you undertake to validate risk assumptions?

Constraints

Every project has its constraints and they need to be anticipated and, if possible, resolved. Projects are about problem solving, innovation and being creative. This does not mean cutting corners – in fact, quite the reverse. It means anticipating constraints and identifying strategies to overcome them in order to achieve the desired successful outcome for the project. Unfortunately, not all constraints can be anticipated or assumed, for example escalating costs or unseasonal weather conditions.

Deadlines and the time available

You are likely to have experienced occasional pressure due to deadlines and the time you have available to complete a task, such as completing an assignment or to prepare for an exam. Maybe the time allocated was not managed very well, planning was insufficient or assumptions were not made about unforeseen circumstances, such as falling ill at a time of year when illness is more prevalent.

Reflect

What can you do differently to avoid undue pressure caused by time constraints when completing your course assignments?

Funds for the project, including contingency

Starting up a computing project requires funds, and the budget is often greatly underestimated. For example, what might seem a fairly simple procedure, creating a filter in a database, can take ten hours in a bespoke system.

Therefore a contingency fund needs to be incorporated into the budget to allow for extra time to counteract time lost, for example, because of internet downtime, sickness or lack of technical expertise.

The contingency fund should be set up at the start of the project and the potential risks should be used to identify areas where contingency funds might be needed.

Availability of staff when required

Humans can be unpredictable and unreliable. The project team relies completely on individuals who are committed to the project and willing to be devoted to it. Nevertheless, as humans, we are all vulnerable on occasions, whether due to sickness, accident, bereavement or planned commitments. All these factors can lead to team members being unavailable for project tasks at critical times. The PM will need to assess each individual on their ability to commit mentally and physically and whether they have the capacity in their work schedules to devote adequate time to the project. However, human vulnerability means that there is always a risk that a team member might be unavailable for one of the reasons discussed, so the PM should put in place a contingency plan with respect to team members. For example, they could have other contractors lined up to take over if a team member is going to be unavailable for a significant period.

Reflect

When was the last time you were let down or you let someone down who was relying on you? Was there any way you could have foreseen this happening? What constraints were you or they under? What would you do differently in a similar situation in the future?

Availability of required equipment

Every project requires some materials and equipment and it is the PM's responsibility to check on the availability of these, once the technical team has determined what will be required.

The bakery scenario in Assessment practice 3.1 would require a suitable IT system and software that supports the storage of recipes and the ability to find recipes using a variety of sorts and filters. If Jason does begin writing books, he will require an IT system and connectivity that are stable, along with suitable security and backup facilities, all of which require the correct equipment. The **operating system** or processing power of his current IT system may be inadequate to cope with some IT projects and may need replacing or upgrading.

Key term

Operating system – software that manages computer hardware and software resources.

The PM should plan when particular materials and equipment are needed during the course of the project, as everything may not be needed at the start. They should consider how many of each item of equipment they need, so that they do not over or under order.

Technical expertise in the project team

The PM or project sponsor will want to be confident that the project team have the necessary technical expertise to successful execute the aims of the project. There are many different ways to establish someone's expertise, but the evidence relied on may include:

- observing their practice
- details of their experience on similar projects
- **CV**
- interviewing them
- references
- qualifications
- membership of professional associations.

Key term

CV – stands for curriculum vitae, which is a chronological record of an individual's experience, qualifications and employment details.

Theory into practice

Have you updated your CV lately? Try writing a CV specifically for the role of a technical team member either on an IT networking, website building or computer games designing project. Compare with a peer.

Tip

Remember to include any relevant experience you may have in your CV, whether it is paid work experience, an unpaid internship or voluntary work.

Some of these examples are not necessarily practical but a project sponsor or PM is likely to use a combination of evidence. They will then produce the terms and conditions of the contracts of individual team members, defining their requirements, expectations and termination rules.

If any members of the technical project team have to be replaced during a project it will have a significant impact on the success of the project and, in particular, on meeting the deadlines. Not only is there disruption to the project but also to the harmony of the team itself. The team needs to establish a secure way of working together, which requires trust and respect and when a team member leaves, for whatever reason, the team can become dysfunctional.

Limitations of technology

While technology is a wonderful thing and appears to do just about anything we can imagine (and possibly beyond our scope of imagination), it still has its limitations.

For example, as good as robots are, they are not human and do not (as yet) form relationships or prevent famine, war and disease. Technology, as yet, cannot take over the complete flying of an aircraft, sail a vessel across the high seas or repair the ever-increasing hole in the ozone layer.

However, advancements are being made all the time. Some cutting-edge examples include robots teaching languages in schools, and robotic floor washers, cleaners and lawn mowers.

Link

To find out more about some of these advancements in robotics, see http://www.cnet.com/news/korean-schools-welcome-more-robot-teachers/#!

The limitations of technology can be a constraint on a computing project. The specifications of the computer equipment available may affect the speed with which certain tasks can be performed, and this has an impact on the project schedule. As long as these constraints are identified at the outset of a project, they can be accounted for in the schedule. If the work needs to be carried out faster, then the project sponsor may decide to fund new computers with faster operating systems and more powerful hardware, if they can do so within budget.

Research

Explore how sustainable technology is challenging an 'outdated industrial model' and looking to replace it with eco-innovation. Try searching for 'innovation beyond technology' and also take a look at http://oceanhealth.xprize.org/

The Project Initiation Document (PID)

In the section 'Project management concepts', you were introduced to the notion that every project should have a PID to define the project and state its scope. In this section, you will explore the PID in more detail and examine some examples of templates used in industry.

Production of a PID to contain the key management information

The first pieces of information usually displayed in the PID are key details about the project itself. These include the:

- project title
- project sponsor's name
- project client's name
- project manager's name
- start date
- completion date
- estimated cost.

Document details

The document details are used to control the paperwork associated with the project. What this means is that the details are meant to display any changes made to the PID, and therefore to the nature of the project. The reason for this is that, just as your action plan or individual learning plan will change as you work through your course, so will the project change as it evolves. Every change needs to be documented and an audit trail kept so that all changes can be traced back chronologically.

Every time a change is made, right from the beginning of the initiation of the PID, the details of any modifications will be listed together with the author who made the changes and the date they were made. Each version of the PID generated will be given a number to represent that version. The original is numbered 1.

Table 3.6: Example of document details

Version	Modifications	Author	Date
1	None	J Phillips, PM	7 July 2016
2	Changes made to the deliverables and assumptions	M Fishpool, Project sponsor	21 July 2016

Approvals

Every PID needs to be approved by those in authority before a project can get under way. In the PID, the names, role and dated signature confirming the approval of those in authority on the project are required, not only at the outset of the project idea but also following any subsequent changes to the PID.

This information is displayed on the front of the PID so that anyone involved in the project can immediately see who the sponsors and key stakeholders are and whether the project has full approval to go ahead.

Table 3.7: Example of approvals

Name	Role	Signature	Date	Version
J Phillips	Project sponsor	J Philips	7 July 2016	1
M Fishpool	PM	M Fishpool	7 July 2016	1
A Jarvis	Team manager	A Jarvis	9 July 2016	1

Distribution

Also, towards the beginning of the PID, there should be a list of names of people who should receive copies of the PID. These individuals are the key stakeholders, such as the client, sponsor, PM, any funders, any contractors and members of the project team (or, at least, the team managers). These individuals will be identified by their name and role and a record of when they received a copy of the document and the version they received should also be included.

Table 3.8: Example of distribution

Name	Role	Date of issue	Version
J Phillips	PM	7 July 2016	1
And so on...			

Purpose of the PID

The purpose of the PID is to justify the viability of the project by demonstrating that every aspect of the project has been given careful consideration. The project purpose is represented by the project aims which, in the case of HS2, would read something like 'improve links between the north and south of England' and in the example of the bakery might read 'to implement an IT system which enables the secure storage and easy retrieval of the bakery's recipes'.

Project background, including how the project fits into the organisation

The next item to be included in the PID is the project background, which comprises a description of the project deliverables, the prerequisites required to kick-start the project, how the project will be achieved and the amount of work to be undertaken.

This background forms part of the overall management and control of the project along with the subsequent sections.

Objectives, written as SMART targets

The example shown here in Table 3.9 sets out the objectives for the project with dedicated columns for recording progress or achievement, along with relevant comments and the associated date. All project objectives should be SMART targets, as discussed earlier.

Table 3.9: Example of project objectives

SMART objectives	Progress/achieved	Comments	Date
Create a database of bakery recipes by 1 August 2016	Achieved	Work completed on the database by two members in specified timeframe	2 August 2016
Create an IT system to store and manage the database of bakery recipes by 1 November 2016	In progress	Implementation of IT system underway using plans drawn up by team based on client requirements	23 August 2016

Each of the objectives will represent the stages of how the project aim(s) will be achieved, resulting in what is likely to be a lengthy section of the PID. Nothing can be left to chance, as this document forms the schedule of works for the PM to monitor progress against and for the project team to implement. Each of the objectives will be allocated to team members with the relevant technical expertise required to achieve the objective, all working to the same timeframe and within the same constraints.

Theory into practice

Write out the objectives for the bakery scenario using this or a similar template. You could extend this template to include more detail, such as resources and costs. Ensure that all of the objectives are SMART.

Scope – a statement of what is and what is not included in the project

Every PID will include the scope of the project, which is a statement about what is and what is not included in the project. For example, in the HS2 project this will include not just the railway connection but all the infrastructure, the IT systems for communicating with traffic on the lines and sufficient rolling stock to make use of the new rail line. There will be landscaping, rebuilding or refurbishing of roads and the signage for diverting traffic due to road closures.

Link

For example, the spur to Heathrow airport is currently outside the scope of the HS2 project, although the full scope is still being discussed. Find out more at https://www.gov.uk/government/organisations/high-speed-two-limited

The business case

The business case is the rationale or justification for the project, where the reasons for the project idea are described, the benefits highlighted and options for alternatives might also be given to demonstrate that the thought process has been rigorous.

The business case in the bakery example might suggest an alternative whereby Jason keeps the recipes as they have always been kept but would highlight the risk of the recipes getting lost. Another consideration would be whether Jason could just scan or type the recipes into a digital device using software or an IT system already available. It might be that Jason has not got the skills or the time to do this. A third consideration could be that Jason sends the recipes to a third party who does this work for him, but here the risks to the contents could be far too great, therefore rendering this option as definitely inappropriate.

The business case is likely to include the timeframe for completion and the budget allocated to the project, justified by the anticipated return on the investment (that is, the increase in revenue or savings resulting from a more efficient system). In Jason's case, his time is probably more valuable and better spent baking and perfecting new recipes.

Assumptions

Earlier, you were introduced to the notion that every project will have assumptions made about its execution, for example the impact from risks such as downtime, difficulties with supply and constraints due to bandwidth or processing power.

The PID includes a section where those assumptions are identified and checked for accuracy. The person who **validates** each assumption may justify their integrity by including their **status** and make a brief comment giving reasons for their decision.

Key terms

Validate – this is the process of ensuring that data or systems are accurate by using original sources or tracing sources back to their origin.

Status – (in the context of a business position) this is the role an individual plays or their job title in a professional setting.

Table 3.10: Example of assumptions

Assumption	Validated by	Status	Comments
Client requirements agreed, checked and understood	J Phillips	PM	Jason and PM confirmed specification as matching their requirements
Processing power is inadequate to cope with future intentions. System upgrade included in specification	M Fishpool	Project sponsor	Agreed with client
Etc.			

Constraints

The next section in the PID defines the constraints associated with the project. It is possible, especially with an IT project, that some members of the team will be virtual members. Virtual members could be based anywhere in the world, contributing remotely to the development of the software required for the project. The constraints when working in virtual teams could include:

- managing communication over different time zones, languages and cultures
- ensuring that individuals work as a team despite being in physically distant locations.

Other, more obvious project constraints may include:

- timeframe
- budget
- quality
- availability of technical expertise.

Some constraints may be a combination of all of these factors, for example, whether the individuals with technical expertise are available within the time allocated for the project and within the budget allowed.

Each of the identified constraints also need to be validated. This might be done by the same person who validated the assumptions, depending on their knowledge and expertise.

Table 3.11: Example of constraints

Constraint	Validated by	Status	Comments
If start date is delayed by more than one week, this will conflict with client's annual leave (2 weeks)	J Phillips and Jason	PM and client	Contingency arranged for client sponsor to act as substitute for client
Etc			

Risk management strategy

You are probably realising that project management is a complex process in which numerous features need to be considered before a project can be embarked upon. There are always going to be a number of risks to any project, and IT projects are no exception.

Table 3.12: Example of a risk management strategy

Risk	Probability	Impact	Severity	Contingency Plan
If project sponsor is unable to test system during development	1	Installation delay	Transfer of recipes into system causes minor disruption	Retain copies of all recipes as electronic images and retain all hard copies in current order used by Jason
Etc				

Once the risks have been assessed, the strategies for managing those risks can also be identified. Each risk is identified in the PID along with their likely probability. Some risks will be greater than others but may have a less significant impact on the success of the project than others. The PID template requires each risk to be considered carefully and identified in terms of severity, and the contingency plan to manage each risk should be outlined.

Discussion

In a small group or with a peer, discuss examples of risks you have identified and the strategies you have considered to mitigate those risks. Consider an average day and the potentially risky activities you undertake, such as leaving home, taking a journey and crossing the road.

Theory into practice

What are the risks associated with completing this qualification? Carry out a risk assessment and identify contingencies to ensure a successful outcome.

Here are some of the possible risks associated with an IT system for operating your central heating that uses a simple scale of 0–4, where 4 is the most severe and 0 represents no risk.

Table 3.13: Example of a risk management strategy

Risk	Probability	Impact	Severity	Contingency
Downtime	4	Cold house	Significant in winter	Battery backup
Battery backup fails	2	Cold house	Significant in winter	Manual control
Manual control fails	1	Mostly controlled heating	Usually warm house	Test system in summer time
Run dual system	0	Controlled heating	Warm house	

Link

You can learn more about calculating risk and probability by reading http://www.wikihow.com/Calculate-Relative-Risk

Deliverables

The deliverables are the outcomes represented as recognisable measures, for example the installation of a network or an upgrade of an existing system.

Table 3.14: Example of deliverables

Item	Components	Description
Installation of database	Newly developed and tested database	Recipe database
Installation of imaging software	Picture imaging and editing software	Electronic images of each recipe when produced
Etc		

Each deliverable is shown in Table 3.14 as an item with its components and description identified. For example, an IT project to implement a system for a paraplegic to be able to go about their daily work using a pointer to interact with the user interface might include items such as software, computer unit and pointer.

Project quality strategy

The quality of any project is assessed by whether:

- the deliverables meet the specifications as defined in the PID
- the processes are followed appropriately.

The project quality strategy needs to identify the methods for assuring the quality of the project, for example:

- feedback from the client
- user acceptance testing
- progress made against identified milestones and key targets
- management of the budget to avoid any overspending
- re-evaluating the risks and contingencies for mitigation
- performance of the team and individuals.

Stakeholders

Although project quality is everyone's responsibility, it is a fundamental duty of the team manager, PM and any other key stakeholders. These stakeholders will be identified in the PID with their responsibilities clearly stated, as shown in Table 3.15. This is important because all aspects of project quality need to be covered without unnecessary duplication and, most importantly, without any omissions.

Table 3.15: Example of project quality strategy and stakeholders

Stakeholders	Responsibility
Bakery owner	Validating that outcomes meet project specification
Project sponsor	Monitoring project to ensure that expenditure keeps within budget
Etc	

Research

Examine how other IT projects are quality controlled. You could start by asking the IT department in the place where you study or work. Ask in your Learning Resource Centre for any articles or business books on projects and take a look at websites such as http://www.itplanning.org.vt.edu/pm/qualitymgmtplan.html

Compare your findings with those of a peer.

Representation of the project management team structure as an organisation chart

The PID should also include a visual representation of the project management team structure. This would be a chart on which the roles and, where available, the names of the project management team members are stated.

You may have seen an organisational chart before, perhaps if you have a part-time job or in the place where you study. Figure 3.6 is an example of a project management team structure.

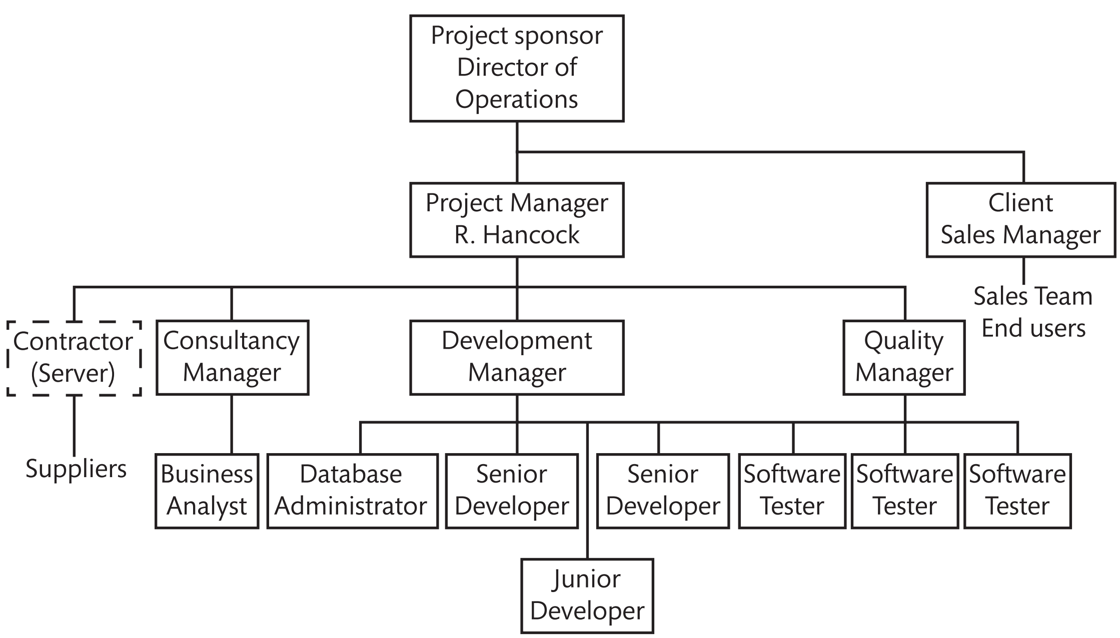


Figure 3.6: Example of project management team structure

Theory into practice

Ask your tutor for a copy of their organisational chart, if you haven't been given one already. Study it and become familiar with job titles and how lines of responsibility are organised. Compare with other organisational charts from external businesses. If you have a business mentor, then perhaps they can help.

Create your own project management team structure for one of the case studies or assessment practice activities in this unit or perhaps for a project you have underway for another unit of this qualification. Seek feedback from a peer or in a small group. You may find that there are several organisational charts in external businesses, divided by departments or line management responsibilities. Ask for a variety so that you can see how they are formed. If you have one from your place of work, it will help you to see the lines of responsibility and authority, and where you fit into the organisation.

Project plan

The project plan is the part of the PID that sets out the finer detail of the project. There are many templates available for this, for example Gantt charts, which act as visual aids for following tasks, responsibilities and timeframes. The overall PID needs to manage the unexpected as well as the expected and one way of spotting whether something in the plan is logistically unsuitable is by producing a visual aid representing the schedule of works. This will enable you to spot obvious clashes and omissions will stand out conspicuously. You may have even noticed this on your timetable if you are studying more than one course at a time.

If you have ever been on a building site and visited the site manager's office, often a temporary building, it is likely that a Gantt chart would be pinned to the wall, identifying these features. You may even have noticed similar charts in the offices or classrooms at your place of study or included in your course induction pack identifying your timetable, assignment or exam schedule.

You might be involved in a project in the future where you have use of specialist project planning software, such as Microsoft® Project or similar.

Research

Explore http://www.projectmanagementdocs.com/project-planning-templates/project-management-plan.html#axzz3zClmdqSJ for examples of an IT project plan and https://www.smartsheet.com/ and http://www.egovernment.tas.gov.au/project_management/supporting_resources/templates for free templates.

Communication plan

A crucial component of any project (and life in general) are the methods and frequency of communication. Consider some of the different methods you can use to communicate, depending on the audience you are communicating with. Communication methods for each project need to be planned very carefully. This will be documented in the communication plan within the PID.

You can probably identify fairly rapidly numerous examples of where communication has been ineffective or non-existent and left you (or someone else) frustrated or confused. The PID includes a section in the communication plan for setting out who, when, how and why to communicate, as shown in Table 3.16.

Table 3.16: Example of communication plan

Stakeholder(s)	Frequency	Type	Purpose
Bakery owner	Daily	Briefings about schedule of events	Keep appraised of progress and what is expected. Plan own schedule to ensure limited disruption
Project sponsor	Weekly	Comparing invoices against specification	Keep within budget agreed
Etc.			

The PM is responsible for taking the lead in planning and organising the communication plan which identifies a schedule for the stakeholders to comply with. Other PID examples will show various templates for communication plans, some of which are very complex. It is up to the PM to determine the amount of detail that is essential for the smooth running of the project.

Some projects will include a series of standardised templates for communicating, such as the structure for emails and a **crib sheet** for telephone calls so that nothing gets overlooked. However, one of the common problems with crib sheets is that they are adhered to regardless of the conversation occurring with the other party. You may have experienced something similar if you have contacted support services such as those provided by remote technical support for software or hardware. Good communication must go beyond a crib sheet, and you must always actually listen to whoever you are communicating with and respond appropriately to what they say.

Key term

Crib sheet – a quick reference checklist of items to cover (in a telephone conversation for example).

Document management

During the life cycle of a project, there might be many changes to the PID and other documents associated with the project, such as plans for resources and financial records. You can imagine how easy it would be to get in a mess with so much paperwork (digital documentation) to control. You might think this seems like a bureaucratic nightmare but a well-thought-through project will still generate a considerable quantity of documentation.

Previously, you were introduced to the idea that the PID indicates the version, author and date of any modifications. When producing documents digitally, a footer can be inserted automatically on each document to identify the version, date, time and **filepath** for the author to officially sign and date. Using these automated features creates an audit trail and makes it easier to locate through a file search. It also means that it eliminates the risk of manual error when updating files.

Key term

Filepath – the term given to the specific location of a document, including the directory in which it is filed.

Changes to documents can also be recorded by tracking those changes within the document so that every single change can be identified. Versions can also be documented during the saving of files in the summary page of the PID, which indicates the modifications made, author and date of change.

This rigorous approach contributes to the management of documents associated with the project. Other ways to management documents are by having procedures for:

- document handling (allocating responsibility to key individuals)
- simple, understandable filing systems
- exchanging documents between stakeholders
- retrieving documents.

Communication and presentation requirements in the PID

In addition to a communication plan, a professional project plan will also include protocols for communication and presentation within the PID. These are a set of rules to ensure consistency and to prevent misunderstandings in the way in which the intentions are construed by the project stakeholders.

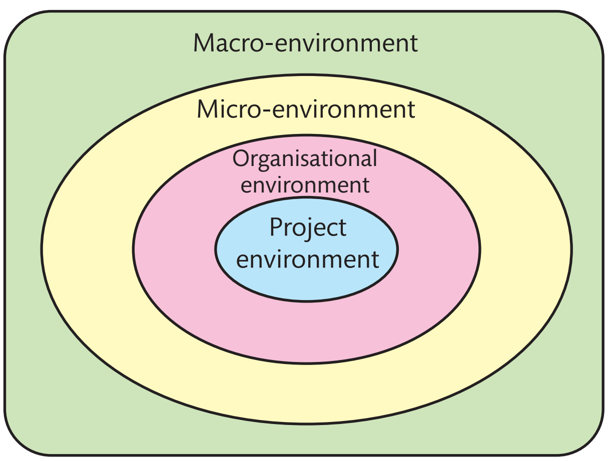
We all have a different view of the world and we interpret it differently in relation to our own experiences and surroundings. This can lead to misunderstandings, assumptions and preconceptions. Consider, for example, a Gantt chart for setting out the schedule of works. If this is presented using a variety of colour coding, it may look attractive but not everyone will be able to interpret it in the same way. Some people have visual impairment which causes them to view colours differently or to find them confusing.

One rule for the presentation of the PID and other project documentation could be to avoid colours and use shading or to use a combination of these. Rules for communication in the PID might include:

- avoid the use of capitals except for names and at the beginning of sentences
- avoid the use of technical language where it might be wrongly interpreted by non-experts
- avoid acronyms unless they are defined in a key (or legend)
- avoid using nicknames or initials for team members as these could be confusing and considered unprofessional
- avoid making assumptions or cutting corners when communicating as these could be misconstrued or cause the project to fail.

Case study

Keeping track of parcels and couriers



The British Computer Society (BCS) conceptualises the model for an IT project planning as shown in the diagram here. BCS identifies the key areas that are vital to the success of any project: the centre circle represents the factors relating to the project and the diagram shows their relationship with the micro (detailed) and macro (bigger picture) factors that influence the overall project design.

In its article 'Case Study of Successful Complex IT Projects' (http://www.bcs.org/upload/pdf/casestudy2.pdf), the BCS describes a business called eCourier, which operates 24/7 in the London area. Parcels are collected from London addresses and despatched to their destinations. As the business has grown, the destinations have extended beyond the local area to global addresses. The business needed an IT system to allocate bookings to the most appropriate vehicle (bike or van) and send a message to the courier to alert them of a waiting job via their handheld terminal.

As well as detecting the nearest vehicle to pick up the item awaiting delivery, the system is also required to gather data on the performance on the couriers and analyse the impact of weather conditions and traffic congestion on the speed at which the courier can carry out the job.

Check your knowledge

- Why does eCourier need a new IT system?
- What benefits is the system likely to bring to the business operations?
- What are the risks associated with introducing a new system?
- How would you define the project mandate?

PAUSE POINT Assuming you were part of the project team for the new eCourier IT system in the case study above, what are the key factors you and your team would need to consider in order to plan this project?

Hint Reread this section and match the facts you can find in the case study against the headings. Identify questions you cannot answer and make assumptions about those you feel able to. If it helps, try creating a spider diagram for planning the project.

Extend Produce a PID identifying the stages in the context of eCourier's requirements.

Assessment practice 3.2

A.01 A.02

Online estate agency 'Wherever I lay my hat...' has been in business for just 4 years and having struggled to reduce their start-up debts are now beginning to reap the rewards, especially as the beginning of 2016 showed a significant increase in the number of properties successfully sold or rented through their service.

Owners Neil and Rashid are looking for ways to improve their service and reduce inefficiencies. They are willing to invest a further £15,000 in their IT systems to integrate software which will enable their property valuers to rapidly and easily produce accurate floor plans that can be uploaded to the website without having to wait for someone to manually draw the image. This will shorten the time taken before the property can be marketed and make savings in terms of personnel.

As both Neil and Rashid have limited IT skills and little experience about initiating a project, they have contacted you so that they can present something official to the bank that will be lending the funds for the project. The bank manager has asked Neil and Rashid to provide a PID. This will initially be in draft form, but they need a fairly comprehensive starting point that can be developed through further discussions.

Neil and Rashid have asked you to:

- produce an outline PID
- outline the budget for the project
- define the success criteria and project outcomes
- summarise the business case and benefits to convince their bank manager to agree to a loan
- provide a summary in presentation format with notes for Rashid to present to sponsors.

Plan

- What do I know and understand before I begin this task?
- Where will I find other sources of reliable information to help me produce a professional and comprehensive draft PID?
- What skills will I be utilising in this task?

Do

- I know what is being asked of me.
- I can interpret the brief and relate it to what I have been learning about so far.
- I can draw on other experiences related to this topic.
- I understand what I need to do to meet the distinction criteria.

Review

- Where did my strengths lie in this activity?
- What areas do I need to work on to improve further in the short term?
- I can identify what I enjoyed mostly in this task.
- I can identify what I gained most of all from this task, especially when things got really difficult.

Tip

You could begin by reading the full article by the BCS 'Case Study of Successful Complex IT Projects' at http://www.bcs.org/upload/pdf/casestudy2.pdf

Next, you could try searching for examples of project planning at:

- https://www.projectsmart.co.uk/
- https://www.mindtools.com/pages/article/newPPM_85.htm
- http://www.formbirds.com/free-project-initiation-document-template

PAUSE POINT What could you do to provide Neil and Rashid with a PID suitable for presenting to the bank manager? How can you assist Neil and Rashid to convey the message with their limited understanding of IT and project management?

Hint Explore some of the sources for further reading provided at the end of this unit. Keep notes about the content and usefulness of each one.

Extend Provide a procedure that could be included in the management of the project.

Research

Explore software options for creating floorplans such as http://www.planup.co.uk/

Tip

Procedures are more easily understood when presented in bullet form or represented diagrammatically, for example as a flow chart.

C Project planning

Project planning is the process of creating and updating the plans for a project, to ensure that the project is completed on time, within budget and to specification.

The following sections cover the areas involved in ensuring that a project plan has undergone a robust and rigorous process and that all eventualities, where possible, have been considered prior to implementation.

Scheduling and milestones

You have already been introduced to schedules and milestones as concepts. The schedule of works requires milestones against which to measure progress just as you have shorter term targets in action plans.

An action plan has a target to aim for, just as yours is to achieve your qualification. However, the target is too big and can be overwhelming if considered as one thing. Therefore to plan your project, just as when planning a journey, you need a schedule of how you are going to get there, and on the way you will have markers to see if you are on track, which are the milestones. Each time you reach a milestone and can measure the progress made against it (even if there needs to be some variations on the way as there might be on any journey) you can give yourself a pat on the back and look back on the lessons learned for the future.

To better understand scheduling and setting milestones, assume you have been allocated a project to implement a dashboard for a database. The processes you need to undertake are described below.

Work breakdown structure

The work breakdown structure (WBS) is the breaking down of the project deliverables into components which can be more easily managed in terms of allocating the resources, time and budget required.

Contemplate the questions that immediately emerge about the design, purpose and features of the database dashboard. You will need to break down each of these into a number of tasks that will need to be included as part of the scheduling. The milestones would represent the stages at which a measurement can be taken to see if you have successfully completed a certain set of tasks and are able to progress to the next stage or whether you need to make changes before moving on.

Task scheduling and precedence, including serial and parallel scheduling of tasks

Each task in the schedule is put into an order defining when it should be executed, that is, there is an order of precedence. This is because some tasks require others to have been completed before they can be started. In the scenario about creating and implementing a dashboard, the database will need to be in place and operational before the dashboard can be designed and implemented. If the project is to network IT systems in a new office building that is under construction, the building would need to be at the **first fix** stage before the cabling can start as it cannot be done unless the basic structure of the building is in place.

Key term

First fix – the term given to all the structural work that takes place to get a building to the stage where cables, pipes and so on are installed. This is before the walls are completed and plastered and the property is completed (known as second fix).

A simple way of showing the order of tasks for the WBS is to number them as a list where each main task is identified by a whole number and each fragment or component relating to the main task is represented as a numerical part, known as 'outline numbering'. For example, one of the main tasks for the networking project is the planning stage, although it will not be the first stage in the project – see Figure 3.7.

2. Planning stage
    2.1. interpret the business case
    2.2. identify the stakeholders
        2.2.3. technical teams
        2.2.4. team managers
        2.2.5. project sponsor
        2.2.6. other stakeholders
            2.2.6.1. identify responsibilities
    2.3. identify assumptions and constraints
3. Produce the PID
    3.1. scope project
    3.2. develop project plan
        3.2.1. Milestone – project plan approval

Figure 3.7: Ordering of tasks for WBS of a networking project planning stage

Tip

Your starting point when undertaking a similar task ordering exercise could be to write or type out all the features of the project and then either number them in order of when they need to occur or cut them up and move them around until they make sense. Number the main tasks with the whole number and allocate the sub-tasks under each one, numbering them when satisfied that they are in the right order. You can always add more later and renumber, although it is easier if you have produced a digital version which will automatically renumber for you.

Another way of presenting the levels and sub-levels for the project is to use a table, as shown in Table 3.17.

Some project tasks will be unique and only carried out once, while others will be carried out several times and possibly at the same time. These are known as parallel tasks.

Parallel scheduling of tasks requires careful planning about the order in which they appear on the schedule – a bit like going on a flight to a foreign country. You will need to show your passport and boarding pass on several occasions to go through security, but you need to queue at the check-in desk only once for your boarding pass before you can move to these stages.

Serial scheduling means scheduling tasks in an order that relies on parallel tasks having been completed. For example, the connection of the cables to a network server can only be undertaken once all the cables are installed and the server is in place to make the connection.

Research

Explore more about task scheduling an IT project by visiting https://msdn.microsoft.com/en-us/library/ff963549.aspx

Discuss with a peer and explore other sources of information on project planning.

Critical path analysis to identify spare capacity in time schedule

A critical path analysis involves analysing when tasks are going to be started and completed, in relation to all the other activities to identify which order they need to appear in on the schedule. The analysis can be undertaken using the critical path method (CPM) which enables a calculation to be made in relation to each of these aspects.

Link

For more about the CPM see http://www.projectcontrolsonline.com/InfoPad/ReferenceZone/PractitionersGuide/PlanningScheduling.aspx

Table 3.17: Extract of table displaying different levels of the WBS

Level 1	Level 2	Level 3	Level 4
2 Planning stage	2.1 interpret the business case 2.2 identify the stakeholders	2.2.3 technical teams 2.2.4 team managers 2.2.5 project sponsor 2.2.6 other stakeholders	2.2.6.1 identify responsibilities

If you think about the networking example used previously, your starting point might be to take these factors into consideration and involve other stakeholders such as the building contractor.

- Determination of the earliest start and finish dates, and the latest start and finish dates of every activity.
  - The cabling is dependent on the builders getting to first fix stage and the finish date will be determined by the start date for the second fix.
- Early dates indicate the earliest start and finish dates when an activity can be performed if all preceding activities have been completed.
  - The early dates in this example are dependent upon the weather, supplies and planning permission and will have an impact on the finish date for first fix.
- Late start and finish dates are the latest dates an activity can be performed without delaying the project.
  - The PM for the building contractor would be undertaking a similar analysis in order to produce their WBS. The PM will need to also identify the latest dates for starting and finishing the project while not exceeding the overall deadline. This is known as the 'longest path'.
  - It is in the interests of each party to complete their tasks within the timeframe, especially when penalties are often imposed by project sponsors. A project may be agreed as a fixed price contract which means that the sooner the project is completed, the more cost efficient the project is for the contractor. Any additional time added to the project is referred to as 'critical path drag'.

Gantt charts as a planning and progress tracking tool

You were learning earlier about how Gantt charts are used as a method for planning and progress tracking. Gantt charts are used to display all the project activities and timeframe regardless of complexity. It might be necessary on a very large and complex project to have several Gantt charts showing different levels of detail but which are all related. Table 3.18 represents a simple Gantt chart template and Figure 3.8 shows an example of Gantt chart software.

Selection and use of project planning software tools

As briefly mentioned earlier in this unit, there are numerous project planning software tools available that provide templates and automated systems to assist with the planning and tracking process. Many of these software tools offer free trial periods to entice you to choose their products. Examples of project planning software tools include:

- ProjectLibre (www.projectlibre.org/)
- SmartSheet (www.smartsheet.com)
- ToDoList (www.abstractspoon.com/)
- GanttProject (www.ganttproject.biz/)
- 2-plan Project Management Software (http://2-plan.com/)
- Open Workbench (http://sourceforge.net/projects/openworkbench/).

Automated software generally makes it simpler to monitor and ensure that all stakeholders are accountable. All relevant stakeholders can have access to the project records and can follow the audit trail of amendments and updates on progress in real time, remotely and 24/7.

Another big advantage of using a software package for project planning is that it provides a procedural checklist, which prompts the PM to consider and respond to each stage in the relevant order to plan the project fully.

The potential disadvantages of using project planning software tools include:

- inaccessible during connectivity downtime
- reliant upon all stakeholders being able to understand and use the system

Table 3.18: Example of a Gantt chart template

	Put a date at the top of each column which could represent days, weeks or longer
	Shade these columns to indicate when the task needs to be undertaken and completed
These rows	
are to list	
different	
activities	
	Write days in the cells above if a task is to be undertaken in parts of weeks

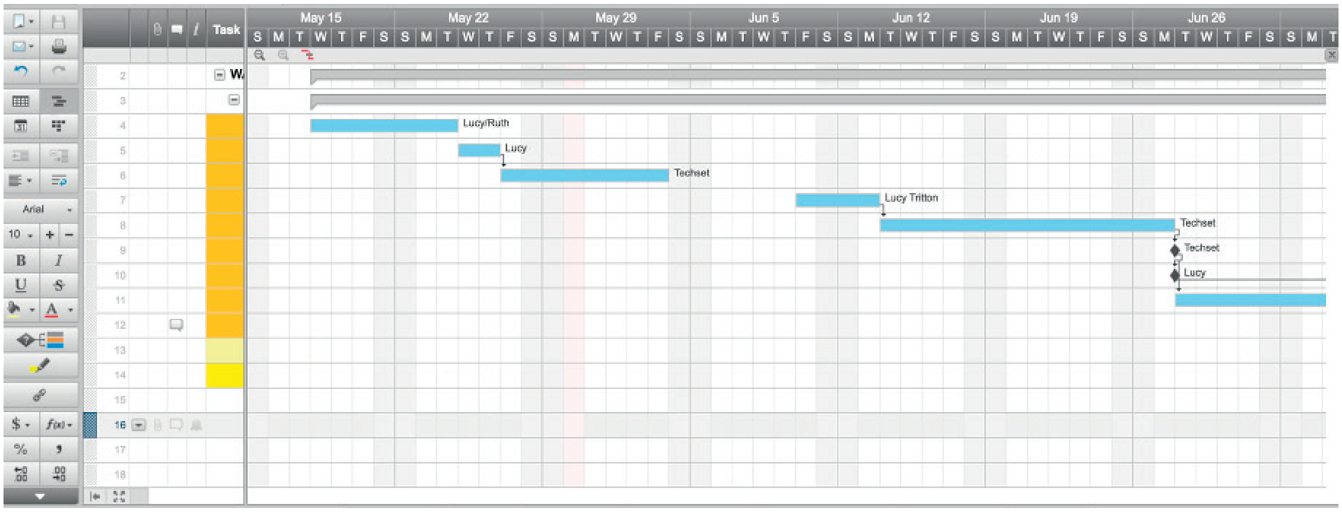


Figure 3.8: Gantt chart produced using Smartsheet

- costs of purchasing software and equipment to operate
- not all project software interfaces or interacts with commonly used Microsoft Office® software
- the cost of training users and time to undertake training
- possible inability of the software to support the different roles of the project team such as PM, team manager, technical team members.

When making a selection about which project planning software tool(s) to choose, you should consider the following factors:

- Simplicity of system – is it too simple or basic to support your project?
- Will the system have longevity – in other words, will the software be suitable for a variety of projects or will other software be required to manage larger, more complex projects?
- If intending to use a bespoke system, how compatible will it be with existing systems and has a maintenance cost been built into the project?
- What ongoing support is available – what if any costs are there and how readily available is the support?

Resources and budgeting

Every project requires resources and, as mentioned previously, a defined budget. It requires considerable skill and knowledge on behalf of the PM to be able to identify all the resources required and to ensure that the calculations for the cost of those resources are within budget without compromising the success of the project.

Resource requirements and allocation

The PM will define the resource requirements and allocate them to each task to be undertaken. How the resources and budget are identified is covered in the following sections.

People and their work allocation, equipment and materials

Every individual working on a project is a resource and will need to be identified and accounted for in order for the PM to calculate their cost and work out how each task will be completed within the timeframe. Each team member will require equipment and materials to execute their work and the PM will need to produce a job description or detailed worksheet for each team member.

The PM works out the resource **logistics** of the project using the project management tool, whether it is software or a manual system (on very small, simple projects). This will give a record of who is doing what, when and what equipment and materials they will be using. Each task will need to be put in order of precedence and although this might appear as simple as putting the tasks in the order you want them completed, there are likely to be **interdependencies**. For example, the logical order to complete a set of tasks might be interrupted because the materials or equipment are not available or because a member of the technical team is required on another project. However, it might be possible for lower precedence tasks to occur instead, until the logical order can be re-established.

Key terms

Logistics – the organisation, coordination and fine detail of complex operations.

Interdependent – two or more things that are reliant (are dependent) upon each other.

Allocation of work and material resources to tasks and pro rata costing

The PM uses the information generated by allocating the work and material resources for each task, to calculate the cost of the project. The cost is likely to be calculated **pro rata** across the task in relation to the technical team, especially where there is more than one individual working on the same task, even if they are required for different periods of time.

Suppose that Table 3.19 identifies three technical team members (TTM) contracted to test a new IT system and the budget for this task is £12,000. The shaded cells indicate the time allocated to each TTM for their involvement in the testing and each TTM is paid £1,500 per week for their time. The pro rata calculation that the PM makes indicates that the project budget would have a **deficit** of £1,500, because, to complete the work, nine weeks worth of work needs to be carried out at £1,500 per week, which is £13,500 (£1,500 over budget). Knowing this, the PM can look at options including increasing the budget or paying the TTMs less per week.

Key terms

Pro rata – in proportion, sometimes calculated as a percentage.

Deficit – financial loss.

The PM would need to re-evaluate the situation to see if the testing could be undertaken in a shorter period, if fewer TTMs could be used or if the PPMs could be contracted at a lower rate of pay. These decisions would need to consider whether the success of the project would be compromised by inadequate testing of the system or whether cheaper labour would result in the project taking longer and costing more anyway if the technical expertise of the TTMs were inadequate.

A major consideration, which could be easily overlooked, is where a pro rata figure has been attributed to each individual involved in a similar task but each is paid a different amount according to their level of expertise or experience. In this case, the pro rata figure would require some adjustment by the PM to ensure that the project does not exceed the budget.

Theory into practice

Problem solving

Imagine you are a PM and have a budget of £16,000 to implement a new network. You have estimated that you need 2 TTMs who are each paid £30 per hour and yourself as PM at £50 per hour. You will all work a 40 hour week. How long do you have for the TTMs to complete the work within budget?

Application of estimation techniques to forecast project duration and cost

To successfully schedule requires a practised skill in **estimating**. The point of estimating is to provide a guideline which you can work from, although that guideline is likely to change as unknown factors become known, just as the estimate for the proposed building of HS2 has already increased from £32 billion to £48 billion.

Key term

Estimating – carrying out a calculated assumption which is not a simple guess but is carefully considered, which results in the best possible approximate answer having taken all known factors into account.

Theory into practice

You, or someone you know, may have asked for an estimate to undertake some work, perhaps a repair on a vehicle, property or equipment. How close was the final bill to the estimate received? If the amounts were different why was that? Were they higher or lower?

How can you ensure that you know the price is fixed before agreeing to the work?

Compare experiences and recommendations with a peer or in a small group.

There are several techniques used to calculate a considered estimate for forecasting the duration of a project and its likely cost.

Table 3.19: Example of a basic pro rata calculation

Human resource	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5
TTM1					
TTM2					
TTM3					

Bottom-up

The bottom-up technique of estimating is thought to be very helpful as it enables a refined approach to estimating. Bottom-up estimating means to:

- break down every task into individual components
- attribute estimates to each individual component
- aggregate the estimates to produce an overall estimate for the entire project.

Link

For more about bottom-up estimating, visit http://project-management-knowledge.com/definitions/b/bottom-up-estimating-technique/

Using this technique is likely to give a fairly accurate or close estimate for the whole project, provided that every component has been considered fully in the analysis. However, it can be an extremely lengthy process especially on larger, complex projects and may take an excessive time to complete, which also comes at a cost.

Parametric testing using simplified function point analysis

Parametric testing is a method used in **statistical** analysis to more accurately predict an outcome. Statistics is a term often misused as referring to numerical data that has not been analysed but just presented as numbers, for example the number of students on your course.

The simplified function point is identified by defining each individual task, often referred to as units, needed to meet the deliverables. These units can be defined by any type of activity, such as:

- take delivery of a new IT system
- unpack the system
- assemble the system
- load the software.

Link

For more about using function points see http://www.softwaremetrics.com/Articles/using.htm.

The method for parametric testing is fairly straightforward and easy to do by following these steps:

1. Identify each unit.
2. Research how many hours are needed for each task.
3. Obtain the best-guess estimate.

Key terms

Parametric – where parameters are assumed so that an analysis can more accurately be estimated.

Statistics – the practice of gathering and analysing large amounts of numerical data such as that used in scientific research.

Link

For more about parametric estimating, see http://4pm.com/parametric-estimating/

Top-down

Top-down analysis is where the PM starts with the overall budget and uses knowledge from similar projects and relates the costs associated with those to the current project. This very simplified approach can be extremely unreliable and has many **caveats** to consider such as:

- inflation
- cost increases for equipment, materials and human resources
- circumstances surrounding the project.

Key term

Caveats – constraints, limitations, conditions.

Research

Read more on estimating by looking at the following websites:

- blog.minitab.org
- http://www.super-business.net/IT-Project-Management/155.html
- http://www.softwaremetrics.com/Articles/using.htm
- http://www.qpmg.com/fp-intro.htm

Budget planning and cash flow to organise resource usage

Some people use a paper based or spreadsheet budget to manage household bills and personal spending. The purpose of budget planning is not only to manage your outgoings but also your incomings. Not considering both sides could result in too much money going out before sufficient funds come in to cover the costs. This is known as cash flow.

Therefore, on a project, the budget planning needs to be sufficiently well planned and organised to manage the cash flow and this is related to how the resource is organised and used.

For example, the PM will need to work out the priority of each resource according to the execution of the task while also considering who will require payment first. If materials are ordered from suppliers and equipment bought or hired, they will need to be paid for, although businesses often manage their cash flow by ordering **on account**. This method enables businesses to begin or continue working despite not having paid their suppliers or having been paid by the client.

Smaller businesses running projects for their customers may request stage payments to ensure that they can continue paying wages and their **creditors**.

Key terms

On account – the term given to the period of time offered to businesses by suppliers before their invoice payments are due to be settled. It is also referred to as 'credit', although, when an individual buys something on credit, they are usually liable for interest payments whereas an 'on account' business arrangement does not incur interest if payment is made within the agreed period.

Creditors – the accounting term given to those to whom a business owes money.

Use of appropriate software tools: spreadsheets and project planning software

Spreadsheet software has may features which are useful when planning and managing projects. The PM or designated member of the project team might be highly adept at using spreadsheets including the higher-level functions, especially those generated by **nested formulae**.

As suggested previously, a PM would decide on the team based on their abilities, experience and expertise. By considering the most detailed and finer aspects of the project, the PM can make savings by utilising the skills that already exist within a team, even if they are not the main reason for contracting a particular individual. A team member with advanced spreadsheet skills may be invaluable to the project, particularly if you are not using project planning software tools. Stakeholders involved in the project will also have skills that could be utilised to support the smooth operation of the project.

However, there is still a case for using specialist project planning software tools rather than just spreadsheets, as the initial work will have already been done for you. The software will come complete with ready-made templates specific to project planning and, although spreadsheets can link together and **populate** cells in worksheets in other files as well as the same file, they may lack the sophistication of the dedicated software.

As project teams may well comprise individuals who have never worked together before or who have not worked in the role assigned to them on this project, using specialist software may well require specialist training. In this case, individuals will take time to become familiar with using the software, which will have an impact on the time and cost allocated to the project. Logistically, bringing contractors together for training and further training to bring them up to speed sufficiently well can be challenging, time consuming and particularly costly. It would be better if contractors were already familiar with the software you are planning to use – in fact your choice of software might be influenced by what your team are familiar with using.

Key terms

Nested formulae – a formula (or several formulae) within a formula, such as 'if...else', used, for example, to calculate the due date of outstanding payments.

Populate – in computing or spreadsheet terms, to input data (automatically or manually) into the application such a worksheet, cell or record of a database.

Risk management strategy

You started to learn about assessing risks associated with projects in the section on 'Project management concepts'. Risk management is a fundamental part of project management. For example, in addition to project budgets being underestimated, software development is often grossly underestimated and could result in the risk of project failure. It is often the maintenance of the software through its lifetime that is underestimated. Suppose that you purchased a computer game or uploaded software to undertake administrative tasks, for example Microsoft® Office. Microsoft® offer support for its products but only for the period they consider to be their lifetimes. Each time Microsoft® releases a later version of its products they withdraw the support from the oldest supported version. This support is also not actually free as it contributes to the software charge and associated licensing.

Risk analysis process

This process is formalised by a series of steps and is described by JISC as 'Risk management is a systematic process of identifying, analysing and responding to project risk'. The steps are:

1 risk identification
2 qualitative risk analysis
3 quantitative risk assessment
4 risk response planning
5 risk monitoring and control.

Link

For more about the five-step-model of risk analysis, see https://www.jisc.ac.uk/guides/risk-management/five-step-model

By identifying the risk, an analysis can be undertaken using contextual information from which a numerical figure can be obtained that identifies the level and priority of the risk. Once these factors are known, the PM (or other stakeholders) can respond to the risk, by planning strategies and contingencies. An agreement will also need to be made on the approach and methods for monitoring and controlling the risk.

Research

Read further about analysing and assessing risks, at https://technet.microsoft.com/en-us/library/cc535373.aspx

Use of impact and probability to calculate severity

Impact and probability are the two main dimensions to use when calculating the severity of a risk. For example, consider the risk of lightning striking you. Although there is a risk, the probability is very low, but, if it were to happen, the impact would be severe.

- Impact (when struck by lightening) – would be severe.
- Probability (of it happening) – is highly unlikely, usually considered to be a 1 in 1,000,000 chance.

Of course it is not that simple, as many other factors need to be considered in this scenario. For example, it will depend on the location and surrounding environment. Also, the ratio between impact and probability will increase or decrease depending on the factors used to calculate the probability.

Use of a risk matrix to classify risks as green, amber or red

In the section 'Project management concepts', you started to learn about assessing the severity of risk by using the 3-point scale. Multiplying the probability by the impact produces the risk to the project and can be presented by classifying risks as green, amber or red.

- Green – ok, everything on track.
- Amber – cause for concern.
- Red – priority concern, danger zone or stop!

You may have heard this referred to as the 'traffic light' system.

You and your teachers might use this simple system to assess any risk of you not completing a task, assignment or assessment activity on schedule. An action plan is then devised to mitigate those risks which is monitored by each of you and updated against the progress being made. As this occurs, the colour coding is changed until it shows green or you have completed the outstanding task.

A more specific way of representing risk using this method is by combining two colours according to their degree of risk, for example amber/red indicates a lack in confidence and a need to monitor closely as there is a greater risk that it could become immediate at any time.

Research

Read more about using traffic lights in projects such as used by the Department of Health IT projects. Try searching http://www.computerweekly.com/news/2240221269/Over-20-government-IT-projects-are-at-serious-risk-according-to-project-watchdog

Contingency planning for major risks

Contingency planning involves the skill of problem solving together with other skills such as analytical thinking and creativity. The example of the risk of being struck by lightning is fairly straightforward. Your options for mitigating the risk of being struck might be:

- researching where the safest place to be is during a thunderstorm or lightning strike (not necessarily indoors)
- avoiding situations of high risk such as sheltering under a solitary tree or standing on high ground
- wearing rubber boots, which are poor conductors of electricity, or sitting in a car

- turning off electrical appliances, avoiding using the telephone, avoiding touching metal objects (so if you are sitting in a car to stay safe, avoiding any metal part of the car) and avoiding using umbrellas or water as they may transfer lightning to your body.

Although this might appear irrelevant, when working on projects that involve building infrastructure, metals or cabling, even the weather can pose a risk. Therefore one suitable contingency may be to plan a project around the intended weather conditions, although this may not be possible or financially viable. However, a contingency in this instance could be to work on other components of the project, possibly remotely or those which are considered to be at very low or insignificant risk during bad weather conditions.

Documenting risks using a standard template

If you have ever seen a risk assessment document produced by a health and safety representative, these can be extremely complex and long documents.

Theory into practice

Ask your teacher for a copy of a risk assessment from the organisation's IT department or health and safety officer.

Review it with a peer and discuss each aspect, what it represents and whether you might have considered the contingencies differently.

Create your own risk assessment for an IT project similar to the implementation of the 2016 fares for TFL (page 136), using a similar template to the example given in Table 3.12 of the section 'Risk assessment strategy'.

Recording issues

It is essential that comprehensive records are kept of project issues. They might appear to be time consuming and overly bureaucratic but, should something major occur, they would be needed to prove who is, and who is not, responsible.

Issues will be recorded in a template or in specialist software to create an audit trail and should be documented by anyone who is working on that aspect of

Table 3.20: Risk matrix – hazard risk assessment

Hazard risk assessment		**HAZARD PROBABILITY**				
		A Frequently	**B Probable**	**C Occasional**	**D Remote**	**E Improbable**
Risk prioritisation matrix		Occurring often: high likelihood of occurring; so controls in place	Repeat occurrences: likely to occur; controls not used or ineffective	History of occurrence: may occur; secondary controls in place	Difficult to occur: unlikely but possible; reliance on work practices to control risk	Very difficult to occur: no history of occurrence; unlikely to occur
HAZARD SEVERITY		5	4	3	2	1
I – Catastrophic: an incident would result in death or uncontrolled environmental impact	5	**EXTREME**				
II – Serious or irreversible: severe occupational injury or illness, or major system damage	4		**HIGH**			
III – Serious or reversible: occupational injury or illness, or major system damage	3					
IV – Marginal: minor occupational injury or illness, or minor environmental impact	2			**MEDIUM**	**LOW**	
V – Negligible: no injury or measurable environmental impact	1					

the project and identifies the issue. Each issue will be dated and the name of the person responsible for logging the issue and the names of any other stakeholders involved will also need to be documented.

Use of an issues log

Recording issues is an important part of the process for gathering lessons learned so that future, similar projects can be planned and managed more effectively and efficiently.

The first time a new task is undertaken it takes about 20 per cent longer than the next time you undertake the same task, which shows that future projects can be more efficient.

Should the risk become a hazard or worse, then, by law, other parties may need to be informed, such as the Health and Safety Executive (HSE), or other government or regulatory bodies.

Link

For more about the HSE, see http://www.hse.gov.uk/healthservices/arrangements.htm

For more about government or other regulatory bodies, see http://www.totalprofessions.com/more-about-professions/regulatory-bodies and http://www.regulatorylaw.co.uk/List_of_regulatory_bodies

Cross referencing to the risk matrix

A risk matrix is a table that combines the risk and hazard with its probability and severity, and which shows the connection between the two as a level. The level of risk is represented by the 'traffic light' system, which is easily recognised and requires little explanation. See Table 3.20 for an example of a risk matrix.

Quality management

You were introduced to the requirement for quality management in the section on 'Project management concepts'. Without systematic quality management processes and systems, a project is likely to fail, resulting in a sub-standard outcome or, at the very least, it will not provide the strong basis from which to gain experience and learn lessons for future projects. In this section, you will learn about the use and application of project quality management processes, techniques and procedures.

Defect removal

Defects can be imperfections, limitations or weaknesses. They can arise as a result of errors made or inefficiencies in the processes used. By keeping an issues log and assessing the risks as an ongoing part of project management, the PM and team can remove any defects as they are discovered or before the project is complete. A defect is usually easy to remedy when diagnosed early, rather than at the end of the project, which is one reason why IT systems are tested before full implementation.

Many software developers rely on customer testing (user acceptance testing) to identify defects even where they undertake testing before offering new products to the general public. For example, Microsoft® and other companies ask for feedback and use that feedback and information gathered about problems with their software to inform changes and developments to software. Updates to apps and service packs associated with software and operating systems include defect removal as well as improvements to the efficiency or features of the programs.

Desk checking and proofreading

As humans, we are always prone to making mistakes – some more than others. Therefore there should always be an element of checking carried out on work done by humans. A check carried out by a human will enhance any checks undertaken by computers. Desk checking and proofreading are two key methods for managing quality and avoiding errors in project management and in all areas of computing.

Desk checking

Desk checking is an exercise carried out by a programmer to check for errors in code that has been written as part of the software development process. They work through the program by hand, often using pen and paper, keeping track of the values of each variable and the statements that are executed. The programmer might choose to only desk check certain parts of the program – in this case, each line of code to be tested is numbered and a record is kept of the result. While a program can be checked by executing it to see if any errors appear when it is running, desk checking enables programmers to spot obvious errors in the writing of the code more easily and ahead of any implementation.

Spell checking and proofreading

The introduction of spell checking features to software has led to a tendency to rely on the spell checker rather than to use our own abilities to check our own work. While spell checkers are designed to check spelling and grammar, they cannot check for meaning. A word could be added by mistake which is spelt correctly and works grammatically but is nonsensical when read, or which changes the meaning from what the author intended. Spell checkers also assume a set language and do not know

colloquialisms. They will not automatically recognise if a word, sentence or even paragraph is missing, if the wrong word has been used or if a name or date is inaccurate. They also do not know what format should be applied to any feature of a document.

Therefore proofreading is required. Proofreading is done by a human rather than a computer and looks not only at spelling and grammar, but also sense. Proofreaders will also check that the styles and formatting that have been set by the author or publisher have been applied correctly. Even in project management documents, such as the PID, it is important to make sure that the information is accurate otherwise a milestone or deadline could be missed because, for example, the document had the wrong date. Obviously proofreading is not infallible as those carrying out a proofread may make mistakes like anyone else. It is particularly difficult to proofread your own work as it is possible to only see what you expect to see and not to spot the errors. This is why it is important that someone other than the author should always proofread any important documents (including those you create for your assignments).

Theory into practice

Work with a peer or in a small group to check pieces of work that you have each produced. How many errors can you spot? Were the errors all similar, and were they due to carelessness or to lack of knowledge?

What strategies will you introduce to avoid making similar errors in the future?

Peer review

Another method for managing quality is by seeking peer review. You may be familiar with the practice of peer review, whereby someone of a similar level of skill and experience assesses another's work against the specification and gives constructive feedback.

On a project, peer review can operate in different ways providing the reviewers operate in an impartial and objective manner. Peer reviewers on projects should be one of the following:

- someone from another technical team
- someone from another project
- a designated stakeholder.

Inspection and walkthrough

Frequent inspections of the processes used and interim testing are another way of managing quality on a computing project. These inspections would be undertaken by designated personnel with the knowledge, understanding and experience to identify any errors or omissions and to test out the project components to ensure that they comply with the specification criteria.

These types of quality checks would be planned in a quality schedule and would include random checks, known as walkthroughs, which are unexpected. The quality checks would be carried out against a set of criteria and the outcomes documented against each criterion.

Testing strategy

Part of quality management is to test the system during construction and before implementation. It is better to identify a problem or mistake before it goes too far towards being finalised as then it becomes more difficult or costly to rectify.

Testing any system requires a strategy for a systematic procedure that is consistent and provides clear guidance on what is to be tested, how is it to be tested, over what period of time and the outcome expected at each stage.

The testing strategy will include a plan that identifies who is responsible and at what stages each component will be tested. Each individual's roles and responsibilities for testing will need to be defined as part of the strategy to ensure that all aspects of testing are covered.

Unit testing against unit specifications

Each unit in the schedule will include a set of specification criteria that identify the intended outcome or deliverables. The unit will be tested against these criteria at the planned stages and a record will be made about what was expected to happen and what actually happened. The testing would be undertaken as part of the quality checking, although this would not only be tested by a designated quality inspector. The personnel constructing the component would also take part in the testing process as they go along.

For example, a software developer writing a program for an operating system would run numerous tests as they produced the code, for example desk checking, so that any errors can be identified in the early stages. If the code produced initially fails to execute the system correctly, then this would need correcting. However, if the programmer did not test the system until the code had been written for the start-up programs, and waited until the program was in the final stages, it would require a considerable investigation to establish why this was the case and to find the error(s).

Integration testing against designs

The testing strategy needs to take into account how the part of the system that is being developed is meant to integrate with other components or units of the system.

This is testing the system for compatibility in terms of how the systems interact with each other and should be compared with the original designs for the system that should specify how the systems should work together. This testing would also check for any conflicts in operation when the systems work together as a group. In other words, assuming that the systems work independently, how do they behave when integrated? Do they behave differently or are there some functions which do not operate? Integration testing also requires checking that the way the system functions is how it was intended to function or appear.

Project testing against the designs specified by the PID, will identify the number of tests to be carried out and how these tests will be undertaken, such as under what conditions and how many times they should be carried out. For example, if the project involved installing wind farms, then testing would involve testing the operation of the wind farms in different weather conditions and measuring how they perform against what is expected. The wind farm equipment should also be tested against the designs prior to construction or by means of a prototype to test the final design prior to full implementation.

You may have heard or read about the testing of Formula 1 race cars in factory wind tunnels where the cars are tested rigorously before being used in races where the risk to human life is significant. These types of tests are known as simulations and are intended to replicate, as near as possible, the real conditions. Simulations are very useful for testing components when the hazards involved if the test were to fail are deemed too great when using other methods.

Research

To read more about software engineering testing, see http://www.informit.com/articles/article.aspx?p=21468&seqNum=6

Systems testing against requirements

Testing a system against requirements is the stage of the testing strategy that involves reviewing the error reports, correcting the errors and undergoing more testing until the team members are confident that any deficits or mistakes have been successfully addressed.

Regression testing

Regression testing is where software is tested for bugs or changes to the features of a program resulting from recent changes in program code. Software development requires considerable diligence and precision to ensure that the code executes the program as expected. Even the slightest change in the code or an omission, such as a bracket or a comma, will affect the operation and can prevent a program from running at all.

Key term

Regression testing – a type of software testing that seeks to uncover new software bugs, known as regressions, in existing functional and non-functional areas of a system after changes, such as enhancements, patches or configuration changes, have been made to them.

Regression testing is required when:

- modifications have been made to a program
- new features have been added
- previous errors have been fixed
- conflicts in a program have been remedied, such as bugs.

Link

You can read more about regression testing at http://www.guru99.com/regression-testing.html

Use of quality standards as an external benchmark

In addition to the quality management strategy created by the project team, there are external quality standards which can be used as an **external benchmark**. Benchmarking provides a baseline to measure against, providing the variables related to the benchmark are similar to those of the project. A well-known saying is 'you cannot compare apples with pears' meaning that, unless they are alike, the benchmark used for measurement will be unreliable.

Link

For more about benchmarks, look back at the section 'Application of current quality standards and subsequent iterations', in 'Quality and deliverables'.

Key term

External benchmark – a baseline measure deemed reliable enough to use as a level for judging whether the performance of an operation is equal to, better or worse than the standard expected.

There are many professional bodies that provide external benchmarks used by government departments to compare standards. For example, the performance between UK industry and other countries is benchmarked by the OECD.

Link

To learn more about the OECD, visit their website http://www.oecd.org/

Internal benchmarks

Internal benchmarking is where results from previous projects are analysed and evaluated to generate numerical statistics that can be measured against subsequent data and analysed for similar projects. Over an agreed period, depending on the frequency of this kind of project, results can be compared to identify trends over time, which provides a benchmark for measuring quality. Benchmarking is also used to measure the milestones that are defined by a target set for deliverables by the predetermined timeframe, such as those stages described in the section 'Application of current quality standards and subsequent iterations', in 'Quality and deliverables'.

The place where you study will gather data about many different aspects of students' performance and experiences. Data will be gathered using a variety of methods, analysed and evaluated to inform what the organisation does well and what it needs to improve on, for example course results, students' satisfaction levels or the variety of extracurricular activities available.

With a computing project, data will be gathered throughout the project, for example from the error log. Managers will analyse data such as the number of occurrences of errors or other issues and the reasons for them. They will compare the results and scrutinise the detail behind the headline data – these are the numbers and the detail that provide the context, such as when, where and possibly why. This scrutiny is referred to as 'drilling down' in the data. Managers will want to know how many times any issue has happened and whether there are any patterns suggesting why this occurred so that lessons can be learned. By drilling down through the data, managers may be able attribute errors to individual team members who may require further training. Data is also analysed to establish what has worked well, just as you and your tutor analyse your progress and achievement on your course. The manager, just as you and your tutor, will want to know what has gone well and why, in order to continue the practice.

Reflect

Read more about benchmarking in projects by visiting http://www.synergybusiness.com/files/PDF/White_Papers/benchmarking_project_management.pdf and http://www.irrodl.org/index.php/irrodl/article/view/221/304

Summarise these documents and compare with a peer.

Communications

Project scheduling will include the identification of appropriate methods for communication within the project team and the frequency for communicating with stakeholders. Having a formal schedule for when, how and where communication will occur provides assurance to stakeholders that they will be kept informed of developments and it allows them to coordinate their own schedules.

Formalising communication methods does not replace the need for less formal communication, for example those occasions when an ad hoc conversation or email are necessary. These kinds of informal communication will be particular useful amongst project team members, alongside their scheduled communications.

Methods for project team communication

The different methods for project team communications that will be included on a project schedule and the frequency of communications with different project stakeholders are discussed in this section.

Meetings and one-to-one discussions

There are multiple ways of communicating with teams and individuals. Two key methods include meetings, usually involving more than two people, and one-to-one discussions, both of which may be held face-to-face or virtually.

The difference between meetings and one-to-one discussions is that the former are formal, planned and involve a number of invited parties. Minutes are usually documented at meetings, which should reflect the salient points and the emergent actions discussed, whereas discussions can often be informal and can take place on an ad hoc basis. One-to-one discussions are not normally minuted as such, but it is useful for those involved to note down the key points and any actions discussed.

Meetings

Meetings are ideally held face-to-face and on-site, especially in the case of construction projects where several key stakeholders need to be included to discuss progress or changes. If possible, it is best to hold meetings face-to-face to allow for body language and to avoid the ambiguities of language that are often introduced when communicating remotely. Virtual meetings are preferable to other remote forms of communication, if communication ambiguities are to be avoided, because, when video conferencing, you can at least see the faces of the people you are communicating with. Virtual meetings are necessary when project team members and stakeholders are geographically far apart.

Virtual meetings can be much more efficient, cost effective and less time consuming than face-to-face meetings, because no one needs to travel from their normal place of work, and

they may not even need to leave their desk. Therefore there are no travel costs to and from meetings and no time wasted in travelling. Virtual meetings are not reliant upon room availability and do not involve the organising of refreshments. All meetings rely on the organiser to ensure that all parties are available and therefore that any time differences have been considered – this is more likely to be an issue where some of the attendees will be remote. Virtual meetings could be considered to be more efficient because team members do not hang around after the end of meetings chatting but, on the other hand, sometimes face-to-face meetings lend themselves to issues being dealt with more quickly because team members can discuss issues immediately after meetings as they leave the room.

The trouble with meetings is they can appear to get in the way of the 'day job'. This occurs when a meeting has not been properly planned and team members arrive unprepared, not knowing what the aims and objectives of the meeting are. This means that they are not able to contribute fully. Meetings should always aim to resolve something and have a clear purpose and direction. Meetings that were scheduled for a particular purpose, but which are no longer needed because the issue has already been resolved, should be cancelled so that more important project work can be carried out.

Meetings usually generate work, as well as aiming to save time by discussing issues with all relevant stakeholders present so that decisions can be taken quickly. This work should be documented as actions in the minutes. Minutes contribute to the project audit trail and are part of the quality management for any project. There should be formal procedures for producing and using the minutes, meaning that each set of minutes should be signed off by relevant stakeholders who confirm that they are an accurate record of the discussion and decisions of the meeting. Any actions resulting from a meeting should be monitored at the next meeting for their status.

Sending meeting requests via email, as is commonly done these days, means that invitations to meetings can be accepted easily and the calendars of attendees can be populated with little effort.

An especially important factor in scheduling meetings is to include the length of meeting (discussion, telephone call or video conference) so that stakeholders can allocate sufficient time and can coordinate their attendance with other commitments around the time slot arranged.

One-to-one discussions

One-to-one discussions should be given the importance and respect attributed to multiple member meetings, in that both participants should come well prepared. Just as with any formal meeting, regardless of membership, one-to-one discussions should be documented, to provide a summary of what was discussed and the decisions and actions agreed.

Memos and notices

Both memos and notices are forms of communication intended to pass important information on quickly. Memos (also known as memoranda) are used less these days as they seem to have been largely replaced by emails, mainly because they can be produced and sent at a rapid pace without relying on administrators to generate memos third hand and then circulate them.

A memo, like an email, should contain a small amount of information or instructions and relate to just one subject. If more than one subject or topic is to be discussed, then it should be documented in a separate memo or email. Both memos and emails should include a subject heading and identify the sender, their position and the receivers, including those copied in for information rather than action.

Notices are another useful form of communication and can be placed in strategic positions around an office to communicate a message to passers-by. Examples include:

- health and safety warnings
- planning notices
- directions
- general updates on progress
- relevant project contact details, such as those of the PM.

Notices are only useful if those they are intended for actually take notice of them. There is a skill in producing notices that get a clear message across without being ambiguous or cluttered and which attract the attention of the intended recipients.

Reflect

How many times have you seen a notice or poster and not actually absorbed its message? Why was that? Which notices have attracted your attention and what was especially effective about those notices that grabbed your attention?

Theory into practice

Produce a notice for your project technical team informing them about changes to the project schedule. You can use your imagination about the changes that are required.

Compare with a peer and each give other constructive feedback that identifies what worked well and what could be improved.

Telephone conversations and video conferences

The communication plan for a project should include a schedule for conversations by phone and/or video conferencing calls. You may think it odd to schedule telephone calls as it is so easy to just pick up the phone and have a chat. But with a project, it is not an ad hoc chat like chatting to your friends or family – it requires organisation to determine the purpose for the call and who the call should be with and when. This is important so that all stakeholders involved know when to expect the call and can therefore organise their own schedules around it. Normally, there will be points they want to raise and decisions to be made. Phone calls are more likely to be used to address urgent matters, which cannot wait for the regularly scheduled team meetings.

Just as with meetings and one-to-one discussions, scheduled phone conversations need to be well planned and adhered to. Imagine the impact of waiting for a call upon which your next task is dependent. What are the consequences if the call does not happen or does not happen on time? It can have a significant impact on you and others because your next task will be delayed and there may be cost implications. Do not forget that time is money and it is also unprofessional to miss a scheduled phone call or meeting. A telephone call is still a kind of meeting and missing a call is no different from turning up late for an appointment. The same is true for video conferences.

Video conferencing (or web conferencing) is an increasingly common method of communicating within organisations. Video conferences save travel costs and time when individuals who need to attend a meeting are located far apart. Creative industries and media companies are especially active in using video conferencing. Being able to hold meetings remotely in this way, makes it possible to work with more geographically dispersed stakeholders, such as contractors, which gives organisations more choice about who they work with.

Quite often, meeting requests will involve a video conferencing option for members attending remotely, while members who are based in the same office might still gather in the same room and link to the video conference together.

Video conferencing and telephone conversations, such as using Webex®, Skype® or Google Hangouts™, also usually require a set of minutes to be produced to confirm what happened and what was agreed. In cases where only a couple of people are involved in a phone call or one-to-one discussion, it is particularly important to employ some means of communication to inform relevant stakeholders of the outcomes and actions of those discussions. Emails, memos or notices may be used for this purpose.

Research

There are lots of video conferencing apps available, some of which are free. Carry out some research into popular video conferencing apps such Google Hangouts™, Skype®, Webex® and Join.Me®. What are the advantages and disadvantages of each app? What other video conferencing apps are available?

Emails and instant messaging

Businesses rely on emails to communicate between project teams and stakeholders. The benefit of using emails is that they can be constructed and sent out any time of the day or night without necessarily interrupting other activities. Emails leave an audit trail of the content, which telephone calls do not unless they are recorded. Emails can provide a record of what is said, and can also be used to record when information is received and read (if certain email settings are used) and so can be used more easily for monitoring events than phone calls.

Emails can be used to circulate updates to stakeholders without always relying on the PM or technical team to deal with general administration.

Tip

The tone of emails can be difficult to get right, particularly if the subject involves any conflicts between team members/stakeholders or discusses things that have gone wrong. Therefore it is important to consider carefully how your email will be interpreted by others. In these circumstances, it is often better to speak directly either on the phone, on a video conference or at a meeting, rather than to send an email, even though it might feel less confrontational and easier to just send an email. Fewer misunderstandings are likely to occur, or they can be quickly resolved, when people speak directly rather than using emails.

Instant messaging is also used as a means of communication and might be used for:

- seeking and giving updates on progress
- confirming arrival times
- arrangements for deliveries
- scheduling unexpected calls or meetings
- rearranging minor events in schedules.

Instant messages should only really be used for things that only require a quick and easy response, not complex

issues, and where an informal approach to resolving minor issues quickly is appropriate.

Online forums, discussion groups and news groups

Online forums are a means of contributing to a conversation by posting messages. Posts can be read and responded to at the reader's convenience. You would use an online forum for non-urgent matter and when you wish to seek out the views of technical specialists, both within your team and externally, about a particular aspect of the project you are working on.

A discussion group involves all the relevant individuals having a simultaneous conversation, during which they can resolve an issue quickly or make a decision about rescheduling a series of events.

A news group is another form of online forum, which involves stakeholders posting messages about a particular topic. Notes about the discussion are then typed onto a central internet site and distributed to all relevant parties using a networking system known as a usenet. News groups are categorised by topic and there are multiple news groups relating to recreational interests such as football, films and music.

Research

Explore the use of a news group and investigate how you would use a similar online forum for a project.

Depending on the type of project and the industry, online forums, discussion groups and news groups are useful forms of communication and are especially prevalent in the technology sector. Using these methods for communicating means that several individuals can be involved simultaneously and they can contribute without interrupting other activities. These forms of communication can all be carried out easily by team members who work remotely from each other.

Collaborative working tools

There are a number of software companies that specialise in producing systems which enable several individuals to work together simultaneously and remotely, using cloud computing and networks. Collaborative working tools enable businesses to connect internally or remotely to hold meetings and liaise with customers and allow teams to work on projects together.

One of the main benefits of collaborative working tools is that they provide a more efficient and productive approach which can also reduce costs significantly, because the project does not incur transport or accommodation costs to bring specialists together to a central location. Examples include Microsoft's® SharePoint and other management information systems (MIS), most of which are paid-for software. Project teams can access and edit the same documents and schedules in real time ensuring that the information they access is current, because as updates are made they are automatically available. These kinds of collaboration tools are freely available, for example via Google™. If everyone in the team has a Gmail™ account, then you can use Google™ tools such as Docs, Sheets and Slides to collaborate on project documents.

Research

Explore some of the more sophisticated collaborative tools available and find out what they do. You could start by looking at http://www.techworks.ca/services/59-collaborative-tools.

The term 'collaborative working tools' also refers to any system that enables individuals to come together. This includes tools that you are very likely to have used, including instant messaging and video/web conferencing tools such as Skype®, FaceTime® and Google Hangouts™.

Reflect

What methods of communication do you use most often? Are these the most effective methods of communication? What other methods could you use and how would that be likely to affect the success of your communications?

Devising a communication plan

The communication plan shown in Table 3.21 will identify a schedule for communicating with stakeholders by detailing the minimum communication requirements.

Link

Look back at the example of a communication plan in Table 3.16 in the section 'Communication plan' in 'Production of a PID to contain the key management information'.

The information in the communication plan provides stakeholders with the frequency of meetings (and other forms of regular communication) which could, for example, be a standard **briefing** or **debriefing**. A briefing will provide relevant stakeholders, such as the technical team, with a reminder or expansion on the task set out in the project plan.

Key terms

Briefing – an explanation of what is going to happen, such as the order of tasks or where team members are located.

Debriefing – an event where relevant stakeholders reflect on how the task or activities went in relation to the outcome and whether they met the deliverables.

Debriefings are a useful exercise to help team members learn from mistakes and celebrate what went well. Debriefings are similar to when your teacher asks you how a particular task or activity went, for example reflecting on a recent exam or assessment experience.

Communication plans should also include the purpose of the scheduled communication and more detailed plans will include the dates, times and length of each communication.

Frequency of communication

It is important to identify the frequency of communication, not only so that stakeholders can schedule and plan accordingly but also to provide a sense of teamwork and involvement.

Some PMs hold briefing meetings with their team every day before the start of work not only to give information, but as a means of monitoring progress and identifying any emerging issues before they escalate. Debriefings may also be held frequently as they provide an opportunity for the team to reflect on performance and make improvements to processes during the course of the project.

Target audience

The term 'target audience' refers to those people involved in the communication. You may have heard, or even experienced, a situation where a meeting or communication has taken place and you have wondered why you were involved and what role you were expected to play. Any form of communication should involve those who are necessary to assist in a successful outcome, whether its purpose is to solve problems, make decisions or amend and update schedules.

If someone does not have the relevant skills, experience or role to participate in the activities of a meeting or other communication, then they should not be invited to participate. For example, junior members of a team should not always be invited to senior management meetings where high-level decisions are being made. However, these same team members will be needed to contribute in weekly team meetings or daily catch-ups, because the work they do relates directly to the purpose of the meeting or conversation. The communication plan should clearly differentiate between those who need to be directly involved and those who are indirectly involved, for example those who need to know but do not need to perform any action. If the objective of the communication is to agree on a change to the central server from the one specified in the original plan, then not every stakeholder needs to be involved unless it impacts on other decisions already made and agreed upon, such as the budget. In this example, the budget holder would need to be involved but other team members might just be copied into any communication to keep them informed until a decision is made.

Agendas and minutes

You have been learning about the use and importance of keeping meeting minutes and also about planning and preparing for meetings. The normal way for communicating the purpose of meetings to those involved is by preparing an agenda. An agenda will include a summary of what will be discussed in the scheduled meeting as well as a list of who should be in attendance, as shown in Figure 3.9.

Agenda: meeting title

Where/how the meeting is to be held

The time the meeting will start and should end (it may finish earlier)

Who will chair the meeting and who will take minutes

List of attendees (this might also indicate who was invited but will be absent)

Agenda items (usually in this order):

1. welcome, introductions and apologies (for non-attendees)
2. meeting objectives
3. review of minutes from previous meeting and confirmation of accuracy
4. outstanding actions from previous meeting minutes
5. topics for today (associated with whoever will lead each topic)
6. any other business (AOB)
7. date, time and place of next meeting, thank you and close.

Figure 3.9: Example of a meeting agenda

The agenda can be fairly standardised, in other words, the same format and template can be used for every meeting, which is helpful because stakeholders will become familiar with what to expect and it provides them with a feeling of professionalism and effective organisation. A routine procedure should be followed for the distribution of the agenda to the attendees so that they can prepare in good time and have all relevant information available for the meeting.

It is fairly common practice to issue papers with an agenda, and, prior to the meeting, a request for these (known as a call for papers) will be sent to relevant stakeholders (even those who may not attend). They may be asked to prepare and send papers to a designated individual who will coordinate the meeting arrangements and issue the papers at an agreed period before the meeting. This is part of a systematic procedure for gathering topics (the papers) for a meeting and whether this is done depends on the importance and frequency of the meeting. Alternatively, attendees might be asked to contribute topics for discussion at a meeting by adding them to a collaborative online document.

In the meeting, the agenda is worked through and acts rather like a checklist for what should be discussed and when. This applies both to face-to-face and virtual meetings. The agenda identifies who will contribute and lead on each agenda item. If contribution from a particular stakeholder is only needed for certain agenda items, they can be invited to attend only that portion of the meeting. By organising meetings in this way the cost attributed to the meeting (mainly people's time) is kept to a minimum and it also means that sensitive or confidential matters can be discussed with only those stakeholders who need to be involved.

The format for minutes should also be standardised, using a predetermined template which ensures that everything is captured and unnecessary aspects are omitted. Again, this enables stakeholders to become familiar with the structure and layout, and so they will be able to read the minutes more effectively. Figure 3.10 is an example of the things you will find in meeting minutes, but these will vary depending on the industry and purpose of the meeting.

Minutes: Meeting title

Minutes of meetings including key information about the meeting such as:

- time, date and place of the meeting held
- list of attendees and absentees (those who were invited but did not attend)
- structure (which pretty much follows the agenda)
- summary of discussions headed by each agenda item, such as the topic or issue, proposed resolutions, any decisions made and any actions, with a list of those who are responsible for carrying out each action.

Figure 3.10: Example of meeting minutes

Meeting minutes should follow these rules:

- they should include only the key facts
- they should be documented in the meeting by a skilled minute taker, where possible
- actions should be identified even if they do not describe the resolution
- every action should be associated with a time for review or completion and someone should be assigned responsibility for each action
- all actions that are identified must be SMART
- they should identify who is on the distribution list, which may include relevant parties who were involved in the meeting
- they should be produced swiftly after the meeting before the key facts (become somewhat clouded by time) so that actions can be implemented.

The relevant agenda may be attached to the minutes.

Meeting minutes and agendas, like any other form of communication, will need to be distributed by an appropriate means, taking into consideration any sensitive or confidential information.

Theory into practice

Ask your teacher for examples of agendas and minutes, perhaps those from a student forum or perhaps relating to quality monitoring for your course.

Produce an agenda for a meeting relating to a project of your choice. Compare and discuss with a peer or in a small group.

Communication and presentation requirements

The requirements for the presentation of any communication will also need to follow a set of pre-agreed protocols. These protocols are likely to include the formats for both verbal and written communication, such as:

- the use of terminology and language
- the correct means of addressing stakeholders and being addressed yourself as a representative of the business.

Working on a project, especially when you are contracted to work in a team, requires a level of diligence to ensure that you uphold the image and reputation of those who contracted you while also representing the project sponsor.

For example, a government project may involve sub-contracting to a variety of individuals from different companies, each likely to have core values that their company expects them to follow. Yet the government (the contractor who indirectly pays for the services) will also have their expectations, rules and reputation to consider. The PM will need to consider how these different parties can be best managed. When contractors attend site meetings with the sponsor or client, they may be required to wear identification representing the sponsor and introduce themselves as representatives of, or working for, the main sponsor.

Key term

Service level agreement (SLA) – a contract agreed between a service provider and the end user that specifies the standard and level of service expected and identifies how the quality of the service will be monitored and measured and by whom, how and when. SLAs are often used in IT support contracts.

There are other aspects that the PM will need to consider when putting together the communication plan, such as:

- documentation that represents the sponsor or contractor
- accounting protocols such as invoicing and payment procedures
- quality monitoring as dictated in contracts and **service level agreements (SLA)**.

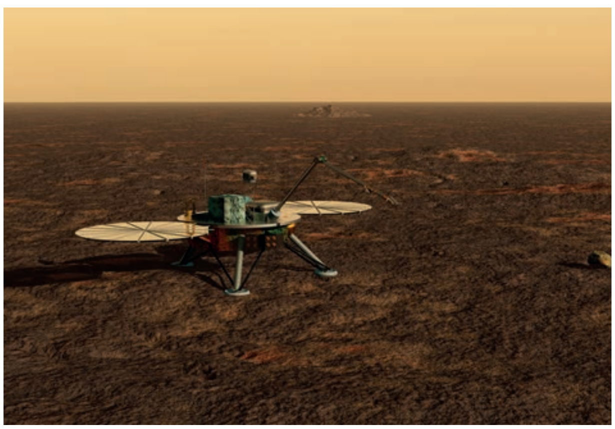
The difference between a PID and an SLA is that the PID relates to the project and the SLA defines the level of standardised services, for example what the customer can expect when purchasing a pair of shoes or receiving an eye test.

Theory into practice

Develop the outline PID that you produced for Assessment practice 3.2 to include a communications strategy and plan.

Case study

The Mars project



In 1999, the Mars Climate Orbiter, which had been launched to study the weather and climate on Mars, was lost when it went out of radio contact with fellow spacecraft Mars Polar Lander. The Mars Climate Orbiter was destroyed when a navigation error caused it to miss its target altitude on Mars by 80 to 90 km and instead entered the Martian atmosphere too low during the orbit insertion manoeuvre. One possible cause for the loss of the space probe loss (that was given in the press at the time) was that two teams involved in the space probe development were using different systems of measurement – one was using metres, centimetres and kilograms, the other was using feet, inches and pounds.

Subsequently, an investigation was initiated into what contributed to the failure of the mission and the findings were issued in a report which identified the main factors contributing to the outcome.

Reports suggested the following.

- The project had failed to set clear success criteria for the project outcomes.
- The scope of the project had not been matched to the funding, leading to inadequate funding for defined outcomes.
- The project was judged to be lacking in good team working and adequate and appropriate staffing.
- The problem was found to be poor communication between, rather than within, different teams.
- The report called for better monitoring of the work of contractors.
- Better training for personnel was required, including the need for project management courses.
- Risk management and issue control was inadequate.

Furthermore:

- any future projects should conduct continuous risk analysis and discussion of issues from the start to the end of the project
- NASA needed to foster a climate in which issues could be easily raised.

Check your knowledge

- What were the ultimate reasons for the failure of this mission?
- What are the lessons learned from analysing the reasons for the loss of the Mars Climate Orbiter?
- What strategies have been recommended as a result of the lessons learned?

PAUSE POINT What are the lessons you have learned from this case study? How do you think they compare with the lessons learned by the project team and other stakeholders?

Hint Research this case study further by reading the report at http://www.dcs.gla.ac.uk/~johnson/Mars/MCO_MIB_Report.pdf

Extend What methods should stakeholders recommend for quality management of a similar project, based on previous and current experience? Form your own suggestions without referring to the report relating to the case study.

Theory into practice

Compare your recommendations regarding the Mars Climate Orbiter with those in the report compiled by the Mishap Investigation Board: http://www.dcs.gla.ac.uk/~johnson/Mars/MCO_MIB_Report.pdf

Initiate a communications plan which could be considered as an effective basis for a similar project.

Assessment practice 3.3

A01 A02 A03 A04

Matthew has been assigned as PM to an IT project for installation of new software and hardware at a local solicitor's office. Matthew has attended several site meetings with solicitor Abdul, a conveyancer, who has raised a number of concerns he has about how the project may affect his business because he has many deadlines to meet in exchanging and completing on property sales and purchases.

As Matthew's business involves many financial transactions, he operates during normal working hours, 9–5 Monday to Friday, but not at weekends. Abdul has made Matthew aware of the importance of ensuring that his clients' information is kept secure and that, if there were any interruptions to the system during a transaction, an entire chain of events could potentially occur whereby a family is left homeless and huge financial penalties would be incurred from mortgage lenders and banks.

Matthew's team comprises himself and two members of his technical team. Abdul, who is funding the project, employs a secretary who manages all the administration for the business and he out-sources all accounting procedures. The installation should take no longer than two weeks. However, Matthew has suggested that his team will need to carry out tests amounting to 2 hours per day. Although the timing of these can be flexible, Fridays should be avoided as this is when the most property transactions take place.

Matthew needs to produce a risk quality management plan to present to Abdul. You have been asked to:

- produce a PID
- identify the quality measures and associated standards together with a testing strategy
- include the scheduling, milestones and risk assessment for the project
- identify the resources and budgeting outline
- provide procedures for evaluating the project.

Plan

- What am I being asked to do?
- How will I present the information?
- What assumptions will I need to make?
- What do I need to research further?

Do

- I am confident that I can identify the salient points.
- I understand the more complex aspects of a task.
- I am able to initiate a plan of action and apply SMART targets to help me achieve a task on time.
- I know what my strengths are and how to improve them even further.

Review

- I can explain how I tackled this task.
- I am confident that I have met the criteria.
- I have performed as best as I can.
- I am proud of my achievements so far.

PAUSE POINT What questions should Matthew be asking next to confirm or refute any assumptions he has made?

Hint Explore some actual case study examples. Perhaps start with https://technet.microsoft.com/en-us/library/cc539247.aspx

Extend Initiate a communication plan for Abdul's secretary as he will be heavily involved in the project. Ensure that you include the relevant stakeholders and other features required.

Discussion

Compare your communications plan with a peer's and share constructive feedback.

D Executing and monitoring a project

No matter how carefully a project is planned, in practice, it may not turn out as expected and plans often need to be changed. These changes may be due to rigorous monitoring and evaluating of the successes and failures of the project and the addition of ways to improve its efficiency and effectiveness. The monitoring of the project throughout its life cycle requires careful and robust planning. This section provides you with an insight into the planning for such processes and introduces you to the challenges that may arise, for which you will need to be prepared.

The waterfall software development life cycle model

The waterfall model is described as the simplest representation of the software development life cycle. The waterfall model of software development is a one way process that is just like water in a waterfall because each stage is followed by another until the process is complete. The waterfall model comprises five stages, as shown in Figure 3.11.

Link

To learn more about the waterfall software development life cycle model, see

http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

http://www.softwaretesting.co.uk/types-of-life-cycle.html

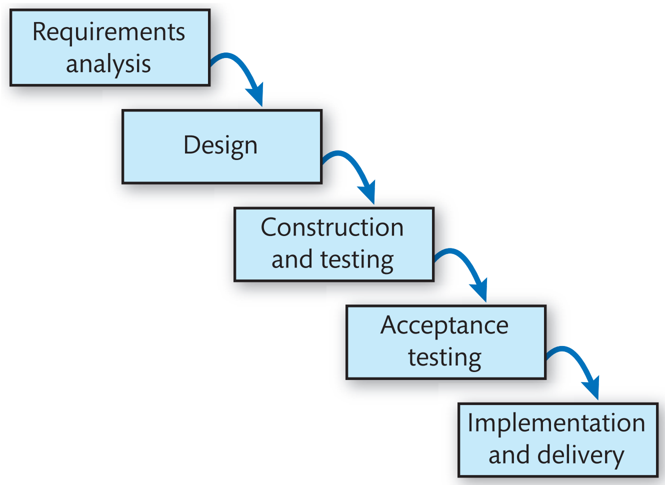


Figure 3.11: Waterfall software development life cycle model

The purpose of displaying the model as a waterfall is to emphasise the relationship between each of the stages and how they are dependent on each other. In other words you cannot design the system if you have not undertaken a requirements analysis. At every stage, a report will need to be produced, which will be **verified** and **validated** by a designated person or persons with the knowledge and experience to confirm that everything has been considered and is viable.

Key terms

Verified – the process of ensuring that data and information are reliable and accurate.

Validated – the process of ensuring that data is valid by using original sources or tracing sources back to their original point.

The advantage of using this model is that it is straightforward and communicates clearly the different stages so that project teams and stakeholders can plan and schedule the project. However, one disadvantage is that the design requirements are needed to initiate a project yet, as you have been learning, these requirements are not always known from the outset.

Each of the five stages might be labelled slightly differently according to the project and its size. For example, the last stage might include 'maintenance' on longer term projects or those which are ongoing. In this section, you will learn about how to use the model to inform the stages of a project plan.

Requirements analysis

The first stage in the waterfall model is the requirements analysis stage in which exactly what is needed to produce a project specification is determined. Therefore, when designing bespoke software for a client, the developer will need to fully understand what the client requires and, as the client is not the technical expert, they are unlikely to know what is achievable or even possible.

This stage is vital to the successful outcome of the project. Every requirement needs to be documented and planned in detail and will need to be tested at each stage. The outcomes of the testing need to be recorded so that they can be tracked and evaluated against the required deliverables. A simple staged model for requirements analysis is shown in Figure 3.12.

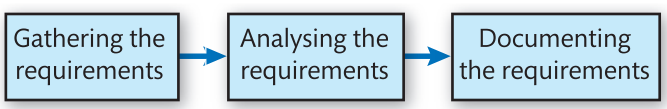


Figure 3.12: Basic staged model for requirements analysis

Depending on any difficulties encountered when gathering the requirements, this model could be represented as a cyclical or iterative approach, rather than a linear approach, whereby the software developer or PM returns to each stage in order to ensure the requirements have been thoroughly explored and defined.

Design

As you will probably have realised by now, the design stage relies completely on a thorough and comprehensive analysis of the requirements. This is the stage where you plan what you are going to design and how you will achieve the outcome the client requires. At this stage there are a number of key factors to include in the planning:

- privacy – permissions for users to know how to access data
- security – features embedded into the design which will verify and validate the data being entered and manipulated
- risk – an analysis of the level of risk at each stage and the strategies for mitigation.

Theory into practice

With a peer or in a group of three, take turns to play the role of PM interviewing a client to establish their requirements for a bespoke software package. Decide upon the scenario before starting. If in a group of three, take turns to observe the role play, making notes about what worked well and what could be improved and why. It would help if you agree a checklist to prompt you as an observer.

When you have each taken a turn, hold a debrief meeting to discuss each of the role plays, making comparisons about the outcomes and generating a list of lessons learned.

Research

Learn more about design by visiting sites such as https://msdn.microsoft.com/en-us/library/windows/desktop/cc307414.aspx

Link

For more about design concepts, see 'Design concepts' in *Unit 4: Software Design and Development Project*.

Construction and testing

This is the stage whereby the software developers construct the system and undertake multiple tests at pre-planned stages to ensure that the program works before progressing to the next stage. It is possible that additional stages of testing will also be undertaken as and when required.

As a PM or part of the team, you will need to have analysed and evaluated thoroughly the stages at which testing needs to be carried out and identified the outcomes expected at each stage. Time for testing and evaluation will need to be built into the project, and this can often be overlooked when allocating budgets. Each test will require a set of criteria to measure the success against and tests will need to be signed off by authorised personnel before moving onto the next stage.

You may be familiar with keeping a log to document events, especially if you are also studying software

development and other practical aspects of computing. As you construct your system or program, you will be following a detailed specification and, as part of a team, you all need to know exactly what stage each team member has reached and is embarking on next, in order to ensure that everyone's work can be integrated simultaneously when expected.

Acceptance testing

Acceptance testing is the stage when the client (including potential users) tests the system and reports on whether or not it meets anticipated requirements. You will need to plan what the client will test, for example whether they will test the whole system or just a unit of it. You will also need to test how the system integrates with other systems already in place. There are likely to be different individuals involved in testing different parts of the system depending on their level of interaction with the system. The people asked to test the system will include relevant project team members and end users. A simple model for this stage is shown in Figure 3.13.

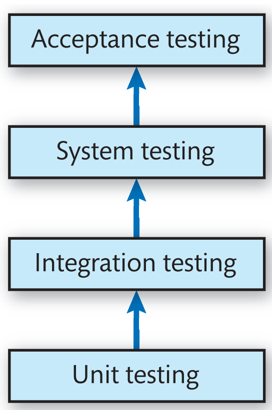


Figure 3.13: Example of model for acceptance testing

As well as the planned testing, this stage is likely to include ad hoc testing and you will need to consider how this can be scheduled while causing the least possible interruption to the flow of the client's business practices, which may have an impact on their productivity.

Theory into practice

Not all testing is to establish whether the complex aspects of the system work (functionality testing) but it is also carried out to establish whether the basic functions are suitable for the client and end users (usability and accessibility testing). Carry out these exercises at http://softwaretestingfundamentals.com/software-testing-exercises/ and compare your results with a peer.

Link

For more about acceptance and other kinds of testing, see 'Testing a mobile app' in *Unit 17: Mobile Apps Development*.

Implementation and delivery

Once the system or program (the deliverables for the project) have been constructed and tested, it is time to implement and deliver them to the client. You may wonder why this is relevant, especially when downloading software can often appear to be a relatively simple process, such as when you download software from Microsoft® or Apple®, or install internet security software. However, with any new software or system, particularly when being implemented across a large organisation, there are many issues to consider and plans need to be put in place to:

- coordinate the implementation date with the client
- arrange delivery methods, including assigning roles to the individuals who will be involved in implementing the new system and specifying what work they will undertake
- giving instructions about timing and any resources or additional equipment needed for the system to **go live**.

Key term

Go live – in the context of implementing an IT system, it means when a product or service is ready for use, i.e. it is active.

- ensure that all documentation is in place for recording acceptance and for raising any unexpected issues and solutions
- provide initial training, where required
- confirm support arrangements.

Monitoring and tracking progress

Earlier in this unit, you started to learn about project quality management and its role in monitoring and tracking the progress of the project and measuring the quality of the output against the specifications.

Project quality management procedures will be formalised and scheduled but there will also need to be random checks to ensure rigour in the quality management process. The findings from each check will be recorded and reported back to the relevant stakeholders along with SMART actions to address any weaknesses or, in more extreme cases, to request a meeting to resolve urgent matters.

Project baseline and variance

The project baseline is used to measure performance by looking at the original scope, cost and schedule for the project. The project baseline will include:

- the deliverables/objectives for the project (i.e. what is being created or set up)
- the budget or planned cost of the project
- the initial start and end dates, i.e. the length of time allocated to the project (which could be represented in hours or days).

These factors can be used to measure the progress of the project and measure performance against the original expectations as the project progresses. The PM, the project team and stakeholders responsible for monitoring and tracking the progress use these baselines to compare with the actual performance.

Variance is the difference between the baseline and actual performance. For example, if the project is scheduled for completion on 7 July 2016 but is actually completed on 21 July 2016, then this is a 14 day variance from the baseline. By tracking progress regularly during a project, it is possible to see where variance is occurring, which gives PMs and project teams an opportunity to put measures in place to reduce it.

Monitoring and recording progress

As explained in the previous section, the PM, the project team and stakeholders are responsible for monitoring and tracking the progress of the project.

One way to do this is to use the 'traffic light' system that you saw earlier. This can be used as a visual aid for indicating how close the project is to the baseline schedule: that is, how much variance there is. This method for tracking progress gives you a quick visual reference to assess whether immediate or short term action is required to stop the project from running over schedule or over budget.

Table 3.21 shows a basic tracking system where each of the traffic light colours is used to provide an instant indication of the progress of the project. Assume it is day 4 of a 22 day project displaying five of the scheduled tasks. The status for each of these five tasks is interpreted for you here.

- Task 1 – started on time and undertaken according to plan.
- Task 2 – started on time and progressing according to plan.
- Task 3 – started slightly late or making slow progress.
- Task 4 – not started yet and may be held up due to slow progress on Task 3.
- Task 5 – not due to start until day 5.

From the simple example in Table 3.21, you can begin to see that slow progress in one task can have an impact on the overall project. This is known as the knock-on effect that each task has on all tasks dependent on it.

As the schedule is monitored and progress is recorded, each activity will be investigated to ensure that it meets the expectations of the baseline in terms of scope and cost, as well as quality, The reasons why any tasks are not being performed as planned will also be investigated. There are likely to be several reasons why a task does not meet the schedule or other expectation, such as:

- the schedule was optimistic
- the deadline was unrealistic
- delays on the supply of materials or equipment

Table 3.21: Example extract of a monitoring and tracking system

Task	**Day**																					
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
Task 1																						
Task 2																						
Task 3																						
Task 4																						
Task 5																						

Legend for example of a monitoring and tracking system	
Dark grey	Timeframe for task to be undertaken and completed
Green and spotted fill	Progress as scheduled
Amber and diagonal lines	Monitor closely
Red and horizontal lines	Urgent attention as running behind

- issues with human resources such as sickness, inefficiencies or technical shortfalls
- environmental conditions
- unexpected occurrences
- delays with tasks on which the current task depends.

Checkpoint reports as a way of recording milestones achieved

You began to learn about the importance of documenting progress and formally reporting on the performance of the project against the milestones that had been set when you first started to learn about quality management earlier in this unit. These reports are known as checkpoint reports and do not just refer to issues but should also provide information about the progress which has met the milestones and where it exceeds expectations. All the information and knowledge gained from the checkpoint reports, often referred to as intelligence, can be used to inform the scheduling of subsequent projects, especially where lessons learned are transferable.

Monitoring risk and managing issues

The skill of delivering a successful project is largely down to the level of monitoring that is undertaken. This depends on the ability of the PM to ensure that when monitoring identifies risks and anticipates potential problems they are managed and mitigated effectively.

If you look back at at Table 3.4, you might question whether the project was sufficiently well monitored and ask why Task 3 is rated amber and Task 4 is rated red (high risk). From this simple Gantt chart, it is impossible to tell why each of the ratings were applied and, importantly, what the significance is of each issue. It might be, for example, that Task 4 will not have a significant impact on the subsequent tasks but that Task 3 could become a major issue. Each of these issues needs to be categorised.

Link

You will learn more about categorising issues in the section 'Categorising issues', in 'Managing issues'.

Recording quality management activity

Each monitoring activity forms part of the quality management strategy, which might also be known as a quality framework. A quality framework is a structure for when and how quality checks will occur and how the outcomes are recorded. These formal accounts will be recorded into templates that provide a checklist of what is required and how those checks are to be performed.

Research

Ask your teacher for a copy of the organisation's quality framework relating to your studies. This framework will show you how the quality of your course is monitored, the frequency of those checks and the methods used.

Carry out research into other quality frameworks and find some templates for documenting quality management activities or an outline for a checklist of quality management activities.

Theory into practice

Discuss with a peer the key features of the framework that could be used to form part of a quality framework for a computing project.

Compare your outcomes with those of another group.

Managing issues

You have been learning about the management of issues and have started to consider how to evaluate issues and their knock-on effect on the rest of the project and specific tasks. In this section we shall be exploring the categorisation of issues, taking action and recording activity as part of the risk management process. The documentation of issues is likely to be undertaken using an issues log. It may be you have used an issues log during your course. An issues log is likely to include the following features:

- the name of the individual who identified the issue
- the date and time the issue was identified
- description of the issue
- the priority for a resolution based on the significance of the issue and its impact on other tasks.

The issues log should also provide an audit trail that identifies any changes to resolutions and should include comments about progress.

Categorising issues

It may be strange to assume that some issues will always arise during the life cycle of a project and that these issues are likely to be unexpected. But this is what makes them issues as, otherwise, suitable alternatives or mitigation plans would have been made at the outset of the project when the original risk assessment was carried out. It is impossible to foresee everything that might occur and it would be unrealistic to assume that you can plan for every eventuality. You can only account for the obvious risks to a project based on the previous experience of the project team and an understanding of the type of project.

Each issue that is identified requires a categorisation to identify both its nature and the level of urgency for finding a resolution. Some of the occurrences of an issue are outlined in this section.

Link

For more about project issue management, see https://www.mindtools.com

Request for change

Requests for change arise because the original planned process for the way a task will be undertaken is no longer viable. For example, it might be that specific materials are no longer available. Another possible example is lack of technical expertise that results in a resolution being proposed which is a change to the original specification.

Requests for changes, also known as change requests, will require permission to be sought, perhaps from the project sponsor or maybe, on a smaller scale, the PM. Associated documentation explaining the request for change would need to be provided to justify the reasons for the change.

Off-specification

Sometimes the only way to resolve an issue is to go off-specification, that is, to alter the original specification of the project. Any instances where it is necessary to go off-specification will need to be agreed with stakeholders. The associated documentation would need to provide assurance that there was no alternative, or that the proposal was the most suitable alternative.

Imagine how challenging it could be for the technical team to convey an alternative solution to stakeholders who do not have technical expertise. While the issues log and other records might provide a clear rationale to those involved in executing the project, it is highly likely that the PM, team leader or an appropriate team member, will need to provide a written report to key decision-making stakeholders that sets out the reasons and solutions using non-technical language. Key factors that any stakeholder would be concerned about include the cost, deadline and quality of the expected project deliverables and any such report should explain how the alternative solution affects these factors.

Problems and concerns

Problems and concerns may arise that are not necessarily categorised as issues, such as the concerns raised by the traffic light rating shown in Table 3.4. Unexpected problems may arise that are even more difficult to manage than those associated with technical issues, such as:

- customers experiencing difficulties managing the change to usual working practices
- conflicts arising between team members
- difficulties accessing materials.

Management by exception: reporting unforeseen issues to the project sponsor and the potential impact on the project

Issues usually occur unexpectedly (i.e. they are unforeseen) and dealing with these issues can have a significant impact on the schedule. This means that the project may encounter several unexpected issues simultaneously which will interrupt productivity.

These unforeseen issues will need to be reported to the project sponsor. This is known as management by exception, that is, you only escalate unforeseen issues to project sponsors and key stakeholders. The PM will report these unforeseen issues to the project sponsor to keep them informed about the potential impact they might have on the project as a whole and to explain or discuss strategies for dealing with them.

Recording lessons learned

Earlier in this unit, you learned that you should evaluate progress and performance within a project to inform future working practices. This is known as lessons learned. You no doubt have opportunities in your lessons to reflect on what went well and what requires improvement. Both what went well and what did not go so well inform lessons learned.

Evaluating the lessons that have been learned from a project is only half the job. It is pointless to carry out this evaluation if the lessons learned are not formally recorded so that they can be referred to at a later date and applied to future projects. The features of lessons learned that are likely to be recorded include:

- the incident or action
- actual outcome against expected
- the impact
- the conditions and timing (the scope of the task or action)
- an analysis of outcome
- resolution of any issues
- savings or additional costs incurred
- an evaluation of incident or action
- dates and times
- any other factors or features.

By recording sufficient detail, trends and patterns might be identifiable against other records which could inform future practices. Also, if the details are sufficient they should enable other teams and individuals to transfer these lessons to their own procedures and processes.

Change management

Earlier, you were introduced to requests for change. This section explores further the management of project changes triggered by the occurrence of an issue.

Impact on the project

Managing change is possibly one of the most difficult things to do as it can be so unpredictable, especially when associated with human behaviour. Take, for example, those occasions when someone reacts quite unexpectedly to something you have said or done. You might even have found yourself thinking 'I didn't expect that'. You might have found yourself behaving unexpectedly or trying to resist a change that was happening in your life or had an impact on you. To better understand why change can have more of an impact on some people than on others, consider its characteristics in the form of a cycle or wheel, as shown in Figure 3.14.

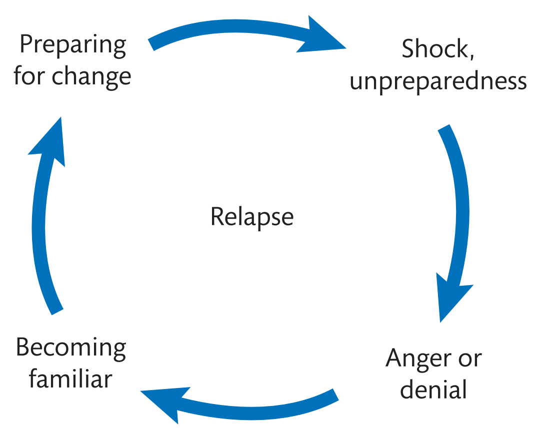


Figure 3.14: Example of why change has an impact on humans

In Figure 3.14, you might view the entry point as the first time someone is exposed to the change, such as bereavement or anything supposedly out of our control. Often this is followed by anger or even denial, then, as we grow more used to the idea, depending on our nature and behaviours, we might become more accepting as we become more familiar and can prepare ourselves for the change. However, we could relapse at any time as other factors have an impact on us.

What Figure 3.14 aims to show is that there are some reasons why people find it difficult to cope with change.

Of course, as we are individuals, when we are put together in teams we can share different views about how we manage change, especially under less familiar circumstances.

Similarly, when working on a project, change can affect us on a small or large scale, resulting in pressures on the budget and ultimately the success of the overall project. Some people cope with change better than others and some people even find change stimulating and enjoy the challenge. Part of the PM's job in relation to the quality management of issues is to evaluate the impact of change on the project to determine the strategies for managing it.

Theory into practice

With a peer or in a small group, each identify a real-life change that is likely to have an impact on you, for example, the building of HS2 or public sector strikes such as those in the National Health Service.
Close your eyes and think about how it affects you. Then make some notes about your reactions and compare them with the stages given in Figure 3.14. Compare with a peer and then carry out the same exercise with each of the examples provided.

What can you do to control your reactions?

Entire project

One of the first things you would want to assess is the degree of impact that the change is likely to have on the project. If it will affect the entire project, then this could be extremely challenging to address. The project team, including stakeholders and PM, will need to be resilient and be able to manage change.

Stage(s) of project

The issue might affect just part of the project but, if more than one stage is affected, the impact could have an effect similar to that of a row of dominoes toppling over, where each one falls on the next.

It is important to know what impact each issue will have on a stage or stages of the project. It is possible that the impact on a particular stage might be greater than the impact ot another issue that affects the whole project.

Change of scope

Depending on the emerging issue, there may be a need to change the scope of the project. Consider a computing project whereby the system for a new software package would operate on PCs only. If the client wanted to change to using an Apple Mac®, the **platform** identified in the original specification might need to be changed and this would have an impact on the scope of the project.

Key term

Platform – can refer to the operating systems used on the system or to the method of delivery, such as cloud based applications.

Requirements and effects on quality

When the scope requires changing, it is not likely to be a simple change of products or systems but usually the entire project would need to be re-evaluated to assess the overall impact of the change. The requirements of the project will be affected by any system alterations and, consequently, the quality management process will also require re-evaluating for its suitability and robustness. Earlier, you were introduced to the idea of a quality framework, which represents some form of scaffolding for processes to build the quality arrangements around. These processes will require timeframes and responsibility for each one will need to be allocated to team members. Other factors, such as the regularity and frequency of testing, will also need to be determined.

Costs

Change management can impact on the costs associated with a project. Any interruptions and changes are likely to incur costs, although the changes are often made in order to reduce costs and improve productivity.

Timescales

It is likely that changes will result in the project taking longer overall. Timescales will need to be assessed and possibly adjusted using SMART principles. If the scope of the project has changed, this might have an impact on the technical teams, according to their areas of expertise and availability. Stakeholders will be affected by changes and timescales are likely to affect their schedules and other arrangements. The benefits of the change need to be balanced against any negative impact they have on timescales, costs and quality. It might be that a change does alter the timescale but actually has a mitigating effect on the deadline compared with what might have occurred if the related issue had not been resolved by this change.

Development changes, handling modifications to designs

Some changes to the project will be development changes, that is, changes to the design of what is being created. These need to be identified as early as possible so that modifications to the designs can be handled. These development changes are likely to involve changes to the deliverables (i.e. to what the client is expecting to receive at the end of the project) but the impact of the changes might extend to the whole project because the schedule, costs and resources might be changed. Handling alterations to designs will involve evaluating the current plan and modifying it to accommodate the changes.

Dealing with faults

Imagine a project where the issues relate to unexpected faults being identified in the IT system, for example TFL's Oyster card issues. The PM would need to call on the technical team to identify where the faults lie and the reasons for the faults so that the issues can be resolved with minimal disruption or cost. This is not always as simple as it sounds and, depending on the type of business, the impact of unexpected issues could even be life threatening. Where faults are diagnosed and the blame is attributed to the project team, there may be a penalty to pay or insurance claims to manage.

Link

To find out more about TFL, look back at the section 'Monitor and control the risks through the project', in 'The risk management cycle'.

Defects in analysis and design documentation

Defects in the analysis stage of the project life cycle can lead to inadequate designs being produced and, if these defects are not identified early enough, the whole project could be implemented before it is realised that the outcome is not what was required. Not identifying defects in the analysis or design documentation has the potential to have a huge impact on the success of a project, because it might therefore go well over budget and be significantly delayed if these defects need to be rectified late on. This is why a very thorough analysis and then, subsequently, a thorough design stage are crucial to the success of projects. If a defect is identified, then a debrief should be carried out to establish where the defects lie. This will give the project team the opportunity to amend the analysis or design and to avoid them recurring.

Software errors

You are very likely to have experienced a software error at some stage, even if only as a user. Detecting errors in software can be problematic as the error may be due to the user, who is likely to be unaware of the preceding processes which led to the unexpected outcome.

Testing the system at planned and random phases during implementation, using a variety of well-thought-through tests which are compared with the expected outcomes, is part of the quality management process. Documenting the actual outcomes of the tests and comparing them with those expected provides analysts with a benchmark and trail for diagnosing unexpected errors.

Link

You can find out more about detecting software errors in *Unit 1: Principles of Computer Science* and *Unit 4: Software Design and Development Project*.

Key term

Algorithm – a process or series of protocols for an automated event, e.g. in computing programming.

Reflect

Identify an occasion when a piece of software did not react as you expected it to. Why was that, was it due to errors in the software or to human error?

How did you deal with the situation and what changes did or could you make to prevent the problem recurring? Share your situation with a peer.

When developing software, errors can be detected using different strategies, including routine tasks such as:

- checking the **algorithm**
- using error detection schemes.

Once the errors in the software are detected, some changes will be required which, even if they are minimal, could have an impact on the schedule and budget.

The change management process

Change management is a process that provides a consistent and standardised approach to managing change in a project team or, more widely, in an organisation. If the PM used a different approach to managing the change on each occasion, this would exacerbate the amount of change to manage. In the following sections, we will look at how the PM can formalise the change management process.

1. Change request submitted by project manager

Having identified the change required, the PM would submit a change request to those identified as key stakeholders. This would be a formal, documented request and, depending on the significance of the impact, may involve the project sponsor and other relevant parties from the management team.

2. Review of the change request by management team

The management team would review the change request in a formal meeting or over several meetings. These meetings would be minuted so that they contribute to the audit trail. If other decisions such as those relating to financial matters are necessary, then each aspect would require the involvement of the relevant decision makers in the team, and all decisions would be recorded.

3. Assessing feasibility of the change of scope

One of the factors the team would consider is whether or not the change to the scope of the project is feasible and an improvement over the current scope. For example, if a software error had occurred following the introduction of an additional feature which was not determined as essential but 'nice to have', it might be that a resolution to fix the software error with this additional feature in place is likely to be too time consuming and costly to warrant the changes. Instead the merely nice-to-have feature should be removed so that the software error is avoided. This will have an impact on time and money but not as much as fixing the software error that has developed.

4. Approval or rejection by management team

Having thoroughly assessed the situation and considered the available options, the management team, which might be one individual, depending on the significance of the issue, will formally approve or reject the change. For example, the team might agree on the change but the decision might be rejected by the finance director. Depending on the outcome of the decision, the PM will need to act accordingly and this may require another solution for managing the issue to be found.

5. Implementation of change by project team

Once a decision has been made, the project team will need to implement the outcome. This may involve changing their schedules or expectations, even if the team believe that there are other or better ways to resolve an issue. Implementing the change will require flexibility, creativity and resilience in the team and skilful handling by those around them (primarily the PM) who are likely to also feel the effects of the change.

Implementation strategy

Having thoroughly planned the products (deliverables) of a project, the PM will agree an implementation strategy with the team. This will involve presenting the client with a number of product delivery options. It is a skill of a successful PM to keep the client sufficiently well informed so that they understand the product delivery options, the reasons for the options and the risks and benefits to the project. The implementation strategy that is agreed upon will normally be one that has the minimal impact on the rest of the project, and the client's business operations and workforce.

Choice dependent on size and complexity of system

The implementation strategy choices faced by the PM and client will depend on the size and complexity of the IT system. For example, a small, independent retailer introducing an IT system for customer transactions might decide to delay the implementation until the system is stable, by retaining the current system while the new one is set up. However, an online distributor relying on automated systems for customer ordering and payment may be faced with more complex decisions, particularly if a software issue may result in a period of time when the system is inoperative. The implementation strategy choices may be as follows.

Direct changeover

Direct changeover occurs when the old system is simply replaced by the new one. Providing the new system has been tested and is fully operational, this might be a better implementation strategy for the independent retailer, otherwise they may be trying to manage two systems in tandem.

Parallel running

Parallel running means running two systems side-by-side or in tandem (both systems are being used simultaneously). There are advantages to this type of strategy, as comparisons can be made with the results from each system, although it does mean that every job has to be undertaken twice, which can be costly and time consuming.

Pilot changeover

A common approach to implementing new systems is to run a pilot or trial of the new system. You may have experienced this implementation strategy when a new system was being introduced in your place of study or perhaps if you have a part-time job.

A pilot also requires careful planning and a timeframe for implementation, testing, reviewing and evaluating. A pilot approach could involve just part of an IT system, allowing time for further development to be carried out while the pilot is running.

Pilot changeovers often involve gathering feedback to evaluate user satisfaction. Lessons can be learned and the system refined before full implementation. However, this can extend the period before full implementation and can also prove costly. In extreme cases, it can result in the system not being implemented at all (however, this may save time and money because implementing the full system and then finding that it is not suitable would be far more costly).

User acceptance testing as part of the quality and review process

Gathering feedback, using a variety of methods, is a useful and effective way for users to test an IT system and for developers to establish whether the users accept the system. This is known as user acceptance testing. This method is also useful for managing change as users are more involved in the process of implementation over a period of time. As they become used to the system, they become better able to identify any issues or areas where they require more training. The user acceptance testing approach is especially effective as part of the quality and review process.

Case study

The case of the Passport Office

The summer of 1999 was an especially hectic and challenging time for the Passport Office due to difficulties encountered with their new IT systems.

The IT project was intended to implement a new, more secure passport system to replace its ageing one, and was being piloted across two of its six sites at Newport and Liverpool. The implementation strategy for the system was planned to cope with the increased demand for passports and would rely on existing staff working overtime and the hiring of casual staff to use the new computerised system. The project took some high-risk decisions without having fall-back plans in place. The Passport Office had not predicted the surge in applications and enquiries and had no means of handling them.

The main issues identified by the National Audit Office following a review of the project failures were summarised as:

- failure to assess and test adequately the time needed by staff to learn and work the new passport processing system, which involved some changes in clerical and administrative processes as well as computerisation
- insufficient contingency planning in the event that implementation might not go as expected
- extending the pilot from one site to another before problems were fully overcome
- failure to communicate effectively to the public.

The impact on the public was significant, with hundreds of thousands of people waiting for their passports and many having to cancel their travel plans. The Passport Office received compensation from contractors, but had to pay out compensation to the public. They recovered the costs by increasing the passport application fee.

Check your knowledge

- In your own words, who was affected by the errors in this project?
- Describe how implementation strategies were used in this project.
- What are the lessons to be learned from this case study?
- Explain the quality management approaches which could have been used to prevent the failure of this project.

PAUSE POINT If you were the PM of a project similar to the Passport Office case study, what change management strategies would you need to consider?

Hint Read more about this case study at https://www.nao.org.uk/report/united-kingdom-passport-agency-the-passport-delays-of-summer-1999/ and https://jiscinfonetcasestudies.pbworks.com/w/page/59388795/Why%20Projects%20Fail%3A%20The%20Passport%20Office

Extend Plan three other implementation strategies for this project, explaining the advantages and disadvantages of each.

E Project closure and post-project review

The closure of a project requires planning just as the project itself does. There will be finalisation stages to organise and processes that ensure a smooth transition for the project team. Following the closure of a project, the team will undertake a post-project review to evaluate the successes and weaknesses of the project and to establish the lessons that can be learned from it.

Closing a live project

Providing that the project scope is clear and the deliverables are SMART, then the completion of the project should be identifiable. However, there are examples where projects continue to run where the end is not so clear. The client may make demands and request additions to the project, some of which may be outside the scope of the project plan. Therefore, just as the project requires clear and definable boundaries, the closure of the project must also be well planned, and completed in an organised and controlled way. Accounts will need to be prepared and eventually settled, to ensure that all materials, equipment and human resources have been paid for.

Research

Read more about project closure at http://www.mastering-project-management.com/project-closure.html.

Moving into the operation and maintenance phase

Once a project has been completed, it will move into operation and may also have maintenance phases. Earlier in this unit, you were alerted to the fact that the maintenance of a system can carry a considerable cost and is the part of a project that often gets overlooked. Although rigorous testing may appear to alleviate the need for maintenance, it is impossible to test a system for every condition and circumstance, no matter how thorough the testing. Also, circumstances change. Upgrades and patches are always going to be needed to any IT system – in particular, security systems because new threats are likely to be identified.

When a database is implemented, maintenance will be required because the software will continue to be developed, for example Microsoft's Access® application. Development of the software may be needed to fix malfunctions or errors which may emerge at any stage and cannot be predicted.

With a building project, there is a period that defines the end known as **snagging**. The snagging period provides a clear end to the project and, once the outstanding items have been attended to, it can be signed off as satisfactory. A snagging period can be applied to computing projects as well. There may also be guarantees provided with aspects of the work, which will vary according to the product or system or its components. If electrical work is involved, then certificates may need to be provided to guarantee that it is safe, and all of these matters will need to be organised as part of the handover to the client.

Key term

Snagging – the process of identifying and reporting outstanding items or areas that require attention in order for a project to be signed off as satisfactory.

Assessing the benefits delivered and planning to review again later

When reviewing the success of the project, you will need to evaluate the benefits delivered to the client against the deliverables expected in accordance with the project plan. This process requires an objective evaluation and it may help to prepare for this by outlining what has been accomplished by the project and using data to support the findings. This evaluation should also address those outcomes that were not so successful and the reasons why.

As the PM, you will probably undertake this task with key stakeholders and representatives from the project team. This will enable you to prepare for a full team debrief where good and less good news can be shared and discussed. Naturally, you should use team evidence gathered throughout the project and the quality management process. The data gathered will be processed, analysed and evaluated throughout the project phases in order to ensure that risks are minimised and issues are managed.

Closing down risk log, issue log and quality log

As the project draws to a close, the logs for risks, issues and quality will also undergo a process of closure. This is to ensure that there is a line drawn between the live project and the operational and maintenance phase. It is likely that the project team will be disbanded at this point, as those involved in the operation and maintenance of projects have different roles to those involved in implementing the project. Operation and maintenance staff are also more likely to be employees of the client.

Each of these logs will need to be analysed, as they will have been throughout the project, to establish whether there are any patterns or trends that emerge so that lessons can be learned. The PM will be responsible for ensuring that these procedures are carried out as part of the quality management process.

Summarising and reviewing the lessons learned

The analyses of the risk, issues and quality logs will be evaluated and summarised. This process can take considerable time, depending on the amount of data being analysed and the trends and patterns that have been

identified. It is possible that the PM will elicit the help of others to evaluate and summarise the findings and to test the validity and authenticity of any conclusions.

The PM will present the summary of outcomes to the team and relevant stakeholders and they will identify the lessons learned. The lessons learned will be used to measure the success of the project and inform the planning of future projects. There will be lessons learned no matter how successful the project outcome.

Review of project success

Once a project has been closed, it is finally time to determine a project's success. This is undertaken in terms of key factors, the SMART objectives and by taking into account the views of stakeholders.

Measuring success requires an objective assessment being measured against a set of predefined key factors.

Whatever methods are used to review project success, they will require a purpose and goal to measure against. For example, the PM will need to identify what success for a particular project looks like in order to ask the right evaluation questions. The questions will need to be structured so that they do not lead or constrain the answers given and it must also be possible to analyse the responses. They should be based around the SMART objectives of the project.

Review of lessons learned

The lessons learned will be used to inform future projects and to analyse how the project may have been planned or managed differently. Each of the measures and controls performed by the quality management systems will be used to inform the overall success of the project and to identify any stages where issues arose and contingencies were effective.

Review project performance against the baseline and project objectives

To review the project performance effectively and objectively, the findings must be measured against the original baseline and the project objectives. Ultimately, the review must be based on evidence gathered to represent the facts rather than opinions.

Each stage of the project will require evaluation against the SMART objectives identified in the original project plan. The project plan will also have identified the baseline for the project in terms of scope, timescales and cost, and the success of the project will be reviewed against this baseline for variance. These measures will include an evaluation of any changes made during the project and therefore involve any amendments that were made to time, costs, resources and scope, and the reasons for those changes.

Research

Read more about project success. Start by visiting http://www.businessperform.com/project-management/project_review.html

Review of final cost, delivery date and quality of product delivered

The cost is possibly the most important aspect of a project because, ultimately, the cost of the project should be less than the amount of money that a company stands to make through its implementation – otherwise it is not a financially viable project. It is often the case that projects end up costing more than the original estimate. If the budget for a project is exceeded, then someone has to pay the additional cost. Sometimes contractors are charged penalty fees if they go over budget or if the planned delivery date is missed, but it is often the client's company that has to foot the bill.

The quality management processes will be used to establish the product quality by comparing the outcomes with the planned deliverables and client expectations. To undertake these evaluations, feedback will be gathered from relevant parties.

Review feedback from key stakeholders

To assess the success of a project, the PM will seek the views of the people affected by the project, the project sponsor, clients, end users and the development team. Each group may view the extent to which the project was successful differently and will focus on different aspects of the project in their reviews.

Sponsor

The sponsor is likely to measure the success of the project in terms of the project objectives being achieved as expected, that is, whether the project is completed on time, within budget and to the expected level of quality.

Clients

The clients are responsible for signing off the project because it was carried out for them. Projects can be delayed if the client is not satisfied with the project outcomes, particularly if there are any outstanding issues that still need to be addressed. Until the project is signed off, it is likely that the client will withhold some or even all of the payment for the project, depending on the terms and conditions originally agreed.

End users

The views of end users are vitally important to the review of the success of a project. They are the people for whom the IT system was created. As they will be using the system, as part of the testing process and when it goes live, their feedback will be invaluable in establishing any downsides or difficulties encountered when operating the system. The quality management process will need to provide opportunities for gathering this feedback even after the project has gone live, just as software developers such as Microsoft® gather feedback through their websites and online through their software. In this example, however, the feedback is gathered as an ongoing process and used to inform updates and amendments which result in service packs and system updates.

Development team

The development team will possibly provide a different view about the success of the project. They will be considering the execution of the project as well as the end product, and will measure success against the milestones and project objectives. The team will no doubt be asked to report on their experience in working on the project, perhaps also commenting on the relationship with the client and stakeholders, where relevant. They will also be measured in terms of their productivity both as a team and as individuals. People who are employed by the project contractor will be managed through an appraisal system whereby individuals are given targets which are reviewed at predetermined points throughout the project life cycle and the year. Some contractors also participate in a formal appraisal system, depending on the policies of the organisation and length of time involved in a project.

The PM will also be expected to provide feedback from their perspective. The PM will be able to see the bigger picture and provide a holistic view whereas team members and stakeholders are likely to report on the aspects of the project in which they were heavily involved. It is these smaller details which often provide useful insights and, when pieced together, can result in major lessons being learned, such as the impact of team dynamics on projects.

Methods to obtain feedback and their advantages and disadvantages

Gathering feedback is crucial for measuring the success of a project, but getting useful feedback can sometimes prove to be problematic. The methods used to obtain feedback require careful planning and each method carries with it advantages and disadvantages in terms of its ease of use and the quality of the feedback obtained.

Interviews

Any interview requires forward planning, and the questions prepared need to represent the outcomes being evaluated. Other factors to consider are who will be interviewed, how the interview will be carried out, where it will take place and when it will happen. Interviews are not always the most appropriate method for obtaining feedback – for example, if the people you need to interview are geographically a large distance away from you. Interviews or any face-to-face meeting take time to carry out, and involves the interviewer and interviewees being available simultaneously. This raises questions regarding:

- availability
- cost
- time
- travel and refreshment arrangements.

Interviewing requires note taking and, where interviews involve several people simultaneously, it can be difficult to be sure that the views being voiced are objective and that everyone gets to have their say. Successful interviews rely very much on the skill of the interviewer and it is important for interviewees to know the purpose and the format of the interview. One of the key factors in the success of interviews is to determine whether or not they are for information giving or information gathering, although most often interviews are information gathering.

It is important that interviews provide opportunities to follow a line of questioning prompted by responses from the interviewees which generate information that may not have been uncovered merely from the prepared questions. Obviously, a balance needs to be struck between getting interviewees to answer the questions you have prepared and need answers to and allowing them to provide other information, as it may not all be relevant. Well-organised and well-attended interviews also provide a good opportunity to gather information **en masse** on the same day and can save time and money, compared with interviewing everyone separately on different days and in different locations.

Key terms

En masse – a group or large body of people.

Questionnaires

Questionnaires might be considered to be a much easier option than interviews. You prepare the questions, post them online, email or hand them out to those you want feedback from and then simply wait for them to

be completed and returned. Questionnaires should provide opportunities for respondents to 'go off-piste', by including comments sections. The data obtained from the comments sections of questionnaires will be **qualitative**, that is, they will be subjective opinions.

The disadvantages of using questionnaires alone are that response rates are likely to be lower and there are no guarantees that the responses are a reliable representation of the stakeholders. Other factors to consider when using questionnaires include that:

- questions might be ambiguous and so the answers might not be what you were expecting
- responses might too limited or might be supplied without context
- the individual completing them cannot ask questions of the person supplying the questionnaire, for example for clarification of a question's meaning
- if used to gather **quantitative** data by selecting one from an uneven number of options, there is a high possibility that the respondent will select the middle option.

Some of the advantages of using questionnaires include that:

- respondents can complete the questionnaires while the PM and team members continue on other tasks
- they are fairly cost effective to produce and distribute
- **quantitative** data can be gathered fairly easily.

Key terms

Qualitative – narrative data such as conversations, text and observations. Qualitative data is difficult to analyse and the results are highly subjective unless verified using other methods to test the views for their reliability.

Quantitative – numerical data which can be analysed and used to present results in different ways, for example percentages or proportions.

Surveys

Surveys are used to gather information, often in less formal surroundings than interviews. You may have been asked to provide your opinions and views for a survey, perhaps when you have been out shopping. Surveys, as with any other method of information gathering, must be strategically planned and the key to an effective survey is that it is carried out in context. For example, you may be asked if you want to try out a new flavour of coffee in a coffee shop or sample a cereal bar in a supermarket. It would be appropriate for a computing project to undertake surveys in the context where the system is implemented, which may provide the opportunity to gather feedback from a variety of sources, including end users as well as other stakeholders.

Surveys can be carried out online or using other methods that do not involve direct interaction, such as by asking questions over the telephone. However, these methods may restrict the quality of the information gathered and can be viewed as intrusive if unexpected.

The advantages of using face-to-face surveys include:

- providing an unthreatening environment for feedback
- they are more likely to gain honest and open opinions
- the experience being fed back on is likely to be current
- opportunities to expand on the questions which arise.

Possible disadvantages of using surveys include that:

- target subjects are too busy to respond, or provide hurried responses
- they require time and skilled use of questions on behalf of the person performing the surveys.

Observation of resulting processes

Observation is a valuable part of gathering feedback. Just as with other methods, carrying out an observation requires planning and preparation and a set of objectives or criteria to define the purpose and categories that are being observed. Questions that should be considered before carrying out an observation include the following:

- What will observation provide that other methods do not?
- Who will be involved and where?
- What information will subjects have about the observation process?
- How will the data gathered be analysed?
- What training has the observer had to ensure that they are unobtrusive and do not compromise the data gathered?

The PM would need to consider carefully whether observation is ethical on a project-by-project basis, and whether the observation process will be **covert** or **overt**. One such example might be observing children's interaction with a new system in a school or nursery. Special permissions would need to be obtained for the protection of the children, and the PM would need to consider carefully the risk and benefit factors associated with using this method.

Key terms

Covert – undercover, such as a 'mystery shopper' who assesses the quality of operations and performance of staff without the staff or shoppers being informed of when (and possibly if) this will take place.

Overt – a completely transparent and open operation where all affected parties are well informed in advance.

The advantages of observations include:

- feedback can be gathered visually while users go about their usual activities
- observations can reveal actions and operations which may not have been discovered using methods where questions are asked
- staff are in their natural environment and interactions between each other may reveal useful feedback
- observations are likely to be more memorable than gathering data using some other methods, especially those where no direct involvement is required, such as questionnaires.

The disadvantages of observations include:

- those being observed can be conscious of their actions and may act very differently to their usual behaviours (not an issue if observation is covert)
- documenting the observations while observing is challenging as actions can be missed
- documenting outcomes after the event compromises the authenticity and reliability of the observed practice
- analysing qualitative data is complex and subjective, requiring topics or themes against which to identify patterns and form conclusions.

Recommendations for future actions based on the outcome of the post-project review

The process of gathering feedback, analysing data and comparing or combining the results with other data gathered can be lengthy and time consuming, but vital for forming a reliable and unbiased evaluation of project success. As Figure 3.15 suggests, the process can be **iterative** or **cyclical**, whereby more data may be required to test results or further analysis of existing data may be necessary to provide a thorough evaluation.

Key term

Iterative – in software development, each part of the development process is separated into small parts and repeated until complete.

Cyclical – a process which is repeated, and goes round and round in sequence.

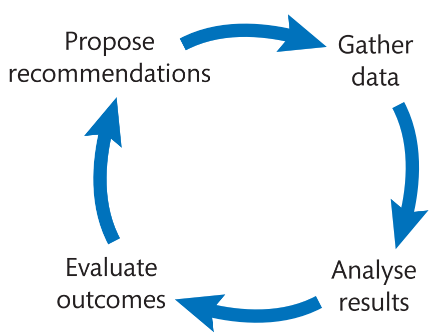


Figure 3.15: Basic example of an analytical process

When the evaluation is complete there will be recommendations for further improvement, development or maintenance. These recommendations may well be time constrained, just as when you discuss your progress with your tutor to review targets and set new ones.

Communication and presentation requirements for reviews

The final stage in a review of project success is to communicate and present the outcomes to the relevant stakeholders. This may be undertaken using a variety of methods, with the feedback and recommendations being communicated to different audiences in different ways, as appropriate.

The PM will need to schedule project review events as swiftly possible to ensure that there is no impact on the closure of the project. Table 3.22 shows some of the likely approaches for communicating and presenting reviews of project outcomes to stakeholders.

The reasons for some stakeholders being involved in more than one event is that the project team includes people involved at different levels, who require different levels of detail. Meetings are likely to include some form of presentation or summarised report providing the information relevant to the audience. Depending on the size of the project and the number of stakeholders, the client and sponsor may or may not attend one or both meetings with other key stakeholders.

For example, the team managers will be interested in the impact on strategic decision making, while the team leaders will be interested in how the strategy affects their operational role. The technical team will require review of the details affecting their activities. The reason for also engaging with them as individuals is that each will receive feedback on their ability to work independently and collaboratively in relation to their performance both as part of a team and as an individual.

Link

To find out more about government-related IT projects, see http://www.parliament.uk/documents/post/pn200.pdf and http://www.instituteforgovernment.org.uk/our-work/more-effective-whitehall/fixing-flaws-government-it.

According to the size and impact of the project, any final reports may be publicly shared, either across a department or the whole organisation: for example, reviews of government-related IT projects are generally publicly announced. This would be a requirement for public spending – the outcomes of such projects will be evaluated by government members and other key stakeholders.

Each method of communicating a review of project outcomes must be meticulously planned and prepared, and the information provided must be accurate, focused and commence with the PM thanking the audience for their involvement with the project. Each episode of communication is likely to close with a summary of what happens next, which may include any scheduled maintenance or follow-up meetings to determine a schedule for progressing recommendations and lessons learned.

The PM will also undergo a form of assessment by the project sponsor and client that will provide feedback on their effectiveness in that role. If the PM is a contractor, then this is likely to be used as evidence for winning future projects.

Table 3.22 Suggested methods of communicating project outcomes to stakeholders

Stakeholder type	Method of communication			
	Face-to-face (group meeting)	**Face-to-face (individual)**	**Statement of outcomes – possibly posted online or sent via email**	**General promotion of project success – website or advertisement**
Key stakeholders including sponsor	✓			
Client		✓		
Team managers	✓	✓		
Team leaders	✓	✓		
Technical team	✓	✓	✓	
Users			✓	✓

Assessment practice 3.4

A01 A02 A03 A04 A05

- You have been asked to step into the PM's shoes due to illness on a project where a fully-integrated computer system is to initially run as a parallel system for 6 weeks to test its stability and effectiveness. The initial development is likely to take around 4 weeks, which equates to 160 hours per team member, and the client wants the project completed within 4 months.
- The PM may return at some stage but you are to initiate the PID and plan the aspects of the project as if you will be PM for its entire life cycle. You will need to ensure that the quality systems are in place, in accordance with ISO/IEC 205010:2010 standards, so that the PM could pick it up readily should they return.
- The project involves the development and implementation of one main database to manage all the business customer records, invoicing and ordering systems. Following a recent assessment by external consultants, it has been discovered that, although there is only a small workforce, every individual likes to run their own version of the database. The sponsor is willing to allocate a budget up to a maximum of £150,000, with £25,000 of that money allocated to any hardware that will be required for implementation.

Available slots

Booking slot ID	Slot Date	Session	Room name	
nnnnn	nn/nn/nn	Xxxxxx	xxxxxxx	Book
nnnnn	nn/nn/nn	Xxxxxx	xxxxxxx	Book
nnnnn	nn/nn/nn	Xxxxxx	xxxxxxx	
nnnnn	nn/nn/nn	Xxxxxx	xxxxxxx	

As a small manufacturing business of cardboard packaging, they have few suppliers of their main raw materials, which are:
- cardboard
- glue and staples
- general office stationery.

The project team will comprise the following.

	Cost per hour	Hours needed for project	Hours required for testing
PM	£40	800	
Team manager	£30	800	
Technical team × 3	£25	2500	120

You have been asked to prepare the PID and plan the project using the templates you are familiar with and any visual aids, identifying the:
- scheduling
- critical path analysis
- quality management processes.

The client has asked you to consider the following.
- What are the issues that are likely to put the project at risk of failure or delay?
- The proposed strategies for managing anticipated issues.
- An implementation strategy.
- How you intend to close the project and measure its success.

Plan
- What are you being asked to do?
- Where will you begin?
- How do you know if you have sufficient budget?
- How will you identify your stakeholder groups?

Do
- I am able to plan out my strategies for undertaking this task.
- I know that I will be able to complete this task on time.
- I am confident that I understand what is being asked of me.
- I am prepared to analyse and evaluate the success of the project using the methods I have identified.

Review
- I am able to demonstrate that I have an in-depth understanding of project planning.
- I am confident that I understand why quality management procedures are so important.
- I can explain the benefits of quality management systems and identify which methods are less useful in this particular project, and why.

PAUSE POINT What aspects of assessment practice 3.4 did you find the most challenging and why?

Hint Review your notes, working and draft documents to recall the moments you found difficult and those you found easier.

Extend What steps do you need to take to further develop your project management skills and how will you get there?

Theory into practice

Seek out the opportunity to undertake a project using the stages you have learned about in this unit. This could be a project you undertake in a small group, and each person could be allocated either different roles or the same role to compare your performances.

Websites

Useful sites for further information:

http://www.britishbuild.org

http://www.cisco.com/

http://computer.howstuffworks.com/

http://www.computerweekly.com/

http://www.dummies.com/

http://www.egovernment.tas.gov.au/__data/assets/pdf_file/0005/78053/Closing_a_Project_Fact_Sheet.pdf

https://books.google.co.uk/

http://www.howtogeek.com/

http://www.ibm.com/

https://www.jisc.ac.uk/full-guide/project-management

http://www.mymanagementguide.com/project-background/

https://www.projectsmart.co.uk/managing-the-project-document.php

https://technet.microsoft.com

THINK FUTURE

Kate McMahon, Software Developer

I've been in my job for four years and absolutely love it. I didn't set out to be a software developer, in fact I really fancied childminding. But then I discovered I really enjoyed coding when I did a taster course in the summer and joined the BTEC National in Computing programme. I had received careers advice and did some searching for future job prospects and that's when I found out about the opportunities that the military had to offer.

So I worked as hard as I could to get the grades I needed as I was told competition is tough and I really needed to go to Uni and get a degree. I'm not saying it was easy and I found the maths hard, but I persevered and it all seemed to get easier somehow. I was thrilled when I was actually offered an apprenticeship with the Army. This meant I not only worked while learning but also got paid for it too!

The experience has been amazing and I sometimes get to travel as well. I am working on lots of projects as part of the technical team and have been told I could expect to lead a small project in the near future. Despite all the doubts I had, I'm now reaping the rewards after what was really a very short period in my life. I'm so glad I tested myself and the prospects are amazing!

Focusing your skills

Balance your efforts

It is important to put in equal effort to all aspects of your study programme – not just the parts you like most. Here are some ideas for you.

1. Create a SWOT analysis so you are more aware of your strengths and the risks of not achieving in your weaker areas.
2. Produce a plan of action which includes some strategies for ensuring that you will achieve in all areas.
3. Identify your short, medium and longer term study goals.
4. Consider your programme as a life cycle project and give yourself rewards when you meet your milestones.
5. Get a poster, screensaver or digital desktop which reminds you of your future career goal.
6. If you are considering a career with the military, visit your local office and explore online websites such as http://www.military.com/veteran-jobs/career-advice/job-hunting/hot-job-how-to-software-developer.html and http://www.army.mod.uk/join/

Preparation

You should also consider preparing for any visits and possible interviews by producing:

- an updated CV
- examples of where you have applied your skills as a student, employee or volunteer, and your social interests and hobbies
- a list of sensible and interesting questions
- a bibliography of the sources you used for research and your application.

Getting ready for assessment

This section has been written to help you to do your best when you take the external examination. Read through it carefully and ask your tutor if there is anything you are not sure about.

About the test

The set task should be carried out under supervised conditions.

- You must not bring anything into the supervised environment or take anything out without your teacher's knowledge and approval. You will need to use a black ink or ball-point pen, so make sure you bring one with you and it might be sensible to have a spare black pen as well.
- You should make sure that you back up your work regularly. You should save your work to your folder using the naming instructions that will be indicated in each activity.
- Don't forget anything else you might need, for example glasses for reading the assessment paper.

Sitting the test

This unit is assessed under supervised conditions and the number of marks for the unit is 66. Pearson sets and marks the task.

The assessment will be carried out in two parts, Part A and Part B. Part A will need to have been completed in preparation for Part B, and Part A and Part B will be submitted together upon completion of Part B.

Part A should be undertaken in 3 hours during the assessment period of one week, which will be timetabled by Pearson. Part A contains material for the completion of the set task under supervised conditions.

Electronic templates for use in the assessment will be provided for centres to download for your use.

The marks for each question will be shown in grey boxes next to each question. Use this as a guide as to how much time to spend on each question.

Practise the project planning as often as you can in preparation for the assessment.

Make sure that you arrive in good time for the assessment and check that you have everything you need for the test ahead of time. Plan out your time during the test to ensure that you allow yourself enough time to complete all the questions and to check through your work at the end. Try to answer every question. Proofread and correct any mistakes before handing in your work. Ensure that you've checked all sides of the assessment paper before starting.

Listen to, and read carefully, any instructions you are given. Lots of marks are often lost through not reading instructions properly and misunderstanding what you are being asked to do.

Key terms typically used in assessment

The following table shows the key terms that will be used consistently in assessments to ensure that you are rewarded for demonstrating the necessary skills.

Please note: the list below will not necessarily be used in every paper/session and is provided for guidance only.

Command or term	Definition
Function point	A way of measuring the amount of work taken to implement part of a software system, for example it might take 10 developer hours to implement a search function.
Gantt chart	A bar chart which provides a graphical illustration of a schedule that helps to plan, coordinate and track all the tasks in a project against a baseline.
Lessons learned	A summary report which brings together any insights gained during a project that can be usefully applied on future projects. This includes factors and actions that supported success and learning from what did not go well.
Modules	Part of a large software system that carries out a specific business role: for example, different departments will use different modules within a full system, e.g. Human Resources will use a payroll module to calculate staff wages. During development, each module is likely to be built and tested independently, often by different groups of developers and testers.
Operating system	The operating system is software that manages computer hardware and software resources and also provides common services for computer programs.
Project kick-off	The official launch of the project; the point at which details of the project are promoted. The kick-off will only happen after some initial investigation to establish that the project is viable, such as: Can the client afford it? Can it be done within the timescale? Is it technically possible?
Regression testing	Regression testing is a type of software testing that seeks to uncover new software bugs, or regressions, in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes have been made to them.
Resource list	A list of all the staff, equipment and raw materials required for a project, along with their associated costs. Staff will usually have an hourly rate or annual salary, while equipment and materials will usually be fixed costs.
Server	Hardware and software that provide centrally managed services on a computer network, such as a database or email system.
Stakeholder	Anyone with an interest in the project. Can include those who have an interest in or can affect or are affected by the computing project. They can be internal or external and at senior or junior level.

Writing long answers

If you are writing a longer answer, try to plan before you start writing. Have a clear idea of the point your answer is making, and make sure that this comes across in everything you write, so that it is focused on answering the question.

- Always make a plan for your answer before you start writing. Sketch this out so that you can refer to it throughout – remember to include an introduction and a conclusion and think about the key points you want to mention in your answer. On this plan, think about setting yourself some timeframes so that you make sure that you have time to cover everything you want to and, importantly (where the question asks for an evaluation), have time to write the conclusion!
- Try to keep your answer as focused on your key points as possible. If you find your answer drifting away from that main point, refer back to your plan.
- Make sure that you understand everything being asked of you in the activity instructions. It might help if you underline or highlight the key terms in the instructions so that you can be sure your answer is clear and focused on exactly what you have been asked to do.

Sample answers

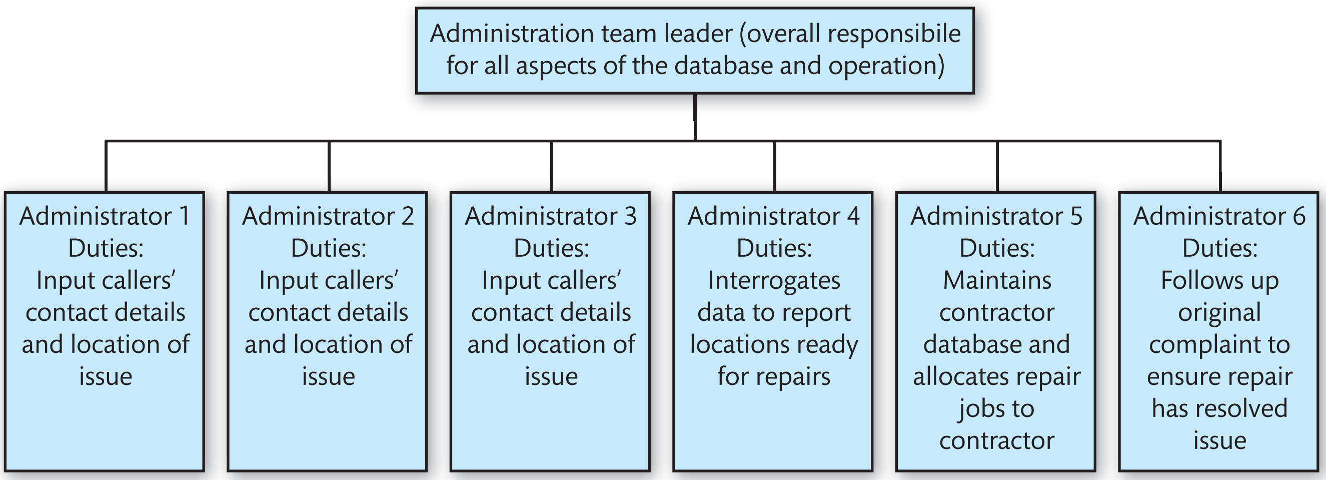
For some of the questions, you will be given some background information on which the questions are based.

Look at the sample questions which follow and our tips on how to answer these well.

Worked example

Set task brief

You are PM of an IT systems project for your local council office that will log all the calls received from members of public who report problems with the roads and pavements in the local area. The information gathered will then be used to organise repairs by maximising the quantity of repairs that can be undertaken in one go by the same contractor. The new database system is ready to be tested using an administrative team of six people.



The team leader is now on holiday for two weeks and there have been delays in getting the system ready for the testing stage. The next stage of the project needs to get underway as soon as possible.

- Produce a risk management strategy using the template provided (see Table 3.12: Example of a risk management strategy, in the section 'Risk management strategy'). Identify the assumptions and constraints using the templates provided (see Table 3.10: Example of assumptions, in the section 'Assumptions' and Table 3.11: Example of constraints, in the section 'Constraints').
- Prepare a schedule for each of the administrators to test the system based around modules.
- Prepare an issues log in preparation for the administrative team to log any issues as they test the system.

For each question you will need to provide a product that relates to the scenario by making reasonable assumptions where the information is not available. You should be able to justify these assumptions and consider alternatives to demonstrate your ability to problem solve.

Write to each of the seven personnel who are affected by the schedule for testing, describing the reasons and what they will need to do. They will also require copies of any logs or recording methods and an explanation of how they are to be used and what to do with this documentation.

With this type of question you will need to consider the audience and their role in the project. This may require you to use different methods of communication and, if you decide that a telephone call was the most appropriate option, you will need to provide a script for the message you are going to give and some form of method for recording the phone call and its contents.

You might find it helpful to put yourself in the shoes of the individual to imagine the types of questions they might have and any confusion they may be encountering about the process. This can help you to consider all angles and to produce a thorough and systematic response. You might need to consider how each person may be affected by the change to their current work practices.

Following the testing, the issues are resolved and the administrative department for highways is operating more smoothly thanks to the new, central recording system. Contractors are able to consolidate their repairs in a more productive and cost effective way and therefore the team are more efficient.

You will need to prepare to close the project and want to gather feedback from the administrative team. To do this you will need to decide on the following:

1 The methods for gathering feedback.
2 The questions you will be asking.
3 How the feedback will be analysed.
4 Justification of the methods chosen.
5 How you will organise these feedback events.

For this type of task, you will need to provide examples of the types of questions you will ask and produce an outline of the template. If, for example, you select observation, the method will still need to demonstrate that you have applied critical thinking to rationalise why you chose this method over others and have considered its shortcomings.

Exam sample answer

The method for gathering feedback from the team leader of the administration team will be by interview. Due to their more senior role, this person will have a holistic view about the overall success of the project. The interview will involve qualitative questions.

As the administrative team comprises six individuals who are communicating with contractors and members of the community, it will be less efficient to involve them in individual interviews and to hold a group interview would mean taking them away from their positions. For those reasons it would be more efficient and cost effective to provide a questionnaire for them to complete.

You and your technical team of three people have analysed and evaluated the project from a technical perspective and, in preparation for receiving all the feedback for analysis, have identified some recommendations for future actions. List those recommendations in a brief report giving reasons for the recommendations and the benefits should they be implemented.

Finally, you will need to arrange the communication and presentation for final closure. Produce an agenda and means of communication to the known stakeholders in this project for presenting the final review.

Software Design and Development Project 4

Getting to know your unit

Assessment
You will be assessed by an externally set task

There is demand within the computing industry for computer programmers who can decompose complex problems and develop solutions. The demand for programmers is so great that not only does it attract generous salaries, but schools are now geared to teach programming to everyone. Problem-solving is a key feature of programming and therefore it is a key skill for software developers. They need to be able to tailor solutions to complex problems in order to develop software that is not only robust, but also meets the requirements of the customer. Within this unit, you will learn to design, test and develop a software solution to a stated problem. You will need to develop innovative solutions to this problem and to evaluate each stage in the software development process.

How you will be assessed

In this unit, you will be externally assessed through a task-based assessment worth 68 marks.

The assessment will assess your ability to work through the software development cycle to produce a program developed in either a C-derived programming language or a version of Python® that meets the necessary requirements. You will design and create the software, design ways of testing the software and carry out the testing once it has been created. Finally, you will evaluate each part of the software development process that you have been through.

You will have six hours supervised time in which to complete the assessment during a one week period, which can be arranged by your centre to take place over a number of sessions. There is no pre-assessment activity and therefore no notes can be used during the assessment.

Grade descriptors

To achieve a grade, a learner is expected to demonstrate these attributes across the essential content of the unit. The principle of best fit will apply when awarding grades.

To pass this unit:

- Learners are able to use their knowledge and understanding of software design and development to design, develop and build software in response to client requirements.
- Learners are able to use standard programming constructs and demonstrate an understanding of how to design and develop a solution with basic functionality which is supported by evidence of testing and evaluation.

To achieve a distinction:

- Learners demonstrate that they can analyse and interpret information related to a given problem and develop a detailed solution to meet all scenario requirements.
- Learners demonstrate an in-depth understanding of programming constructs and show that they fully understand how data is handled within a computer program. They produce an optimised solution which provides accurate results with evidence of thorough testing and evaluation, supported by justification.

Assessment Outcomes

AO1 Demonstrate knowledge and understanding of computer coding paradigms and the software development lifecycle.

AO2 Apply knowledge and understanding of computer coding paradigms and the software development lifecycle to design and create a software product to meet a client brief.

AO3 Analyse information about computing problems and data from test results to optimise the performance of a software solution throughout the development lifecycle.

AO4 Evaluate evidence to make informed judgements about the success of a software product's design and performance.

AO5 Be able to develop a software solution to meet a client brief with appropriate justification.

Getting started

The software development lifecycle involves the separation of software development into different stages. In small groups, consider what the advantages are to breaking the process down into stages. Then write down all the stages you can think of. When you have completed this unit, see how you can improve your list.

A Software development life cycle

The software development **life cycle** is a process that software developers follow. It maps out the life of a software product. The life cycle starts with an idea (sometimes referred to as the 'conception' stage) and moves through each stage of development until it reaches the last stage, known as the evaluation or maintenance stage.

Software developers must follow each stage, which are also sometimes referred to as 'phases'. Each stage produces deliverables required by the next stage in the life cycle process. In this section, you will explore what is involved in each stage of the life cycle and what requirements a software developer must fulfil before they can progress a piece of software to the next stage.

Stages of software development

The software development life cycle is the approach adopted by all programmers when creating software. There are different versions of the software development life cycle. Most software developers follow what is commonly referred to as the 'classic waterfall model'. Developers work through each stage of the process **iteratively,** one after another, until they reach the end.

Key terms

Life cycle – in software development, the cyclical stages required to produce new software.

Iteratively – in software development, each part of the development process is separated into small parts and repeated until complete.

Software development life cycle

The software development life cycle is the name of the overall process of developing software from beginning to end. When starting any new software project, it is important to plan how it will be developed and, once created, how it will be maintained.

Research

The software developers can adopt different approaches or methodologies. One example is the classic waterfall model, but, recently, the 'agile software development model' has come into use by software developers. In small groups, research the agile software development model as well as other methodologies. Consider what the advantages and disadvantages are of each one.

The process is known as a life cycle because it is cyclical, in the sense that, even when a piece of software is initially finished, it might be necessary to go back to the beginning again. This means that a software developer may go through each stage of the process and then repeat it until no more development is required. This repetition is what makes the software development process iterative and is what makes the process a life cycle. For instance, a developer may update the completed software to harness new technologies. As a software developer, you might be able to make substantial changes to how the software works and continue through the development cycle again. Figure 4.1 shows the software development life cycle in diagrammatic form.

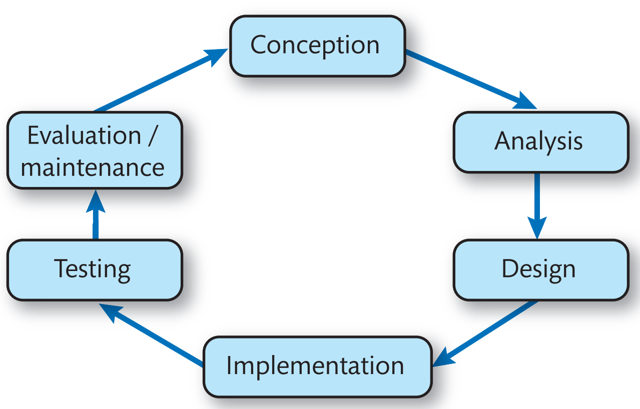


Figure 4.1: The software development life cycle

Conception

The conception stage is the beginning of the life cycle. It usually begins with an idea: that is, what the piece of software is going to be and why it is a good idea to create it. The conception stage is part of the development process where pre-project planning takes place. Pre-project planning involves careful consideration as to whether this is a good business opportunity. There are many ideas for pieces of software, but organisations do not have the time or money to implement them all, so consideration must be given as to whether each idea is viable. Specifically, is this idea for a new piece of software profitable, and how will it benefit your organisation? Does it support your company's position in the market; will it gain you more customers and make money for the company? Does it fit with the philosophy of the organisation? What impact will the development of this software have on staff within the organisation, that is, does the organisation have the human resources to cope with the project? The exploration of these questions should be brief, because not all projects will make the initial cut and it is not sensible to waste time and resources on a project that will not be developed. Essentially, the conception stage is about getting a feel for the potential of a project, and to decide whether it is viable to proceed with it.

Analysis

Once it has been decided that a project will go ahead and the software developed, the analysis stage is needed to identify the organisation's requirements. This stage is the main focus for project managers and stakeholders. Meetings with managers, stake holders and potential users are held in order to determine the requirements of the project – specifically, the requirements of the organisation that wishes to create the software and the requirements of the software itself. These meetings should answer questions such as, 'who is going to use the software?' and 'how will the software be used?' as well as practical questions such as, 'how much money will the software cost to develop?' and 'who will carry out the work?'

It may initially seem that it would be easy to answer these questions but, for a number of reasons, they are often difficult to answer, and are likely to throw up even more questions.

The managers, stakeholders and users may not clearly understand what is required or they may not be able to give sufficient detail to developers.

The user requirements of the organisation may be stated in terms which relate specifically to the business of the client, with which the software developer may not be familiar. For example, a potential user in the banking world will tend to state their requirements in banking and financial terms, but the software developer is an expert in programming, not in banking.

The managers, stakeholders and users may not understand what is (and is not) possible when writing software.

Therefore the organisation's and the users' requirements may need to be the subject of some discussion and negotiation between the client (user) and the programmer. These negotiations eventually culminate in the creation of a 'problem definition statement', whereby all the issues or problems experienced by the client and their users are summarised and details of a software program that could resolve them is laid out. During these discussions, a number of key questions need to be asked.

1 **What are the primary aims of the software system you are going to develop?**
   This is one of the first things that need to be defined and will probably be in terms of the problem that the software is intended to solve or some opportunity it will provide. It is important to clarify this at the start because, sometimes, during the process of developing software, people can lose track of the original reason for which the software was required.

2 **How does the current system work?**
   In many cases, the software to be developed will be replacing an existing system. It may be a manual system (that is, using people rather than computers) or it may be an old computer system that has outlived its usefulness. In any case, it is important to understand the current system thoroughly so that new software can preserve its essential elements and its good features, while avoiding the problems from which it suffers. The new software should be an improvement on any previous system.

3 **What other software systems does it need to interface with?**
   No software system works in isolation. All systems take input, either from the user or another computer system, and also produce some kind of output. Part of understanding how the new software system should be developed involves defining these inputs and outputs.

Research

Research the 'Hawthorne effect'. What is the consequence of this effect when eliciting requirements from clients and the intended users of software?

Design

Once the requirements of the software have been defined, you need to consider how the program will achieve what is required. This is the design stage. You will need to draw up a design for the software program. The design will define the following.

1. The user interface that will be provided. This might include designs for the screens that users will use to input data, and the reports that will be output from the software system.
2. The general structure of the program, including how it will be broken up into procedures and how those procedures relate to each other.
3. The detailed design, showing how each of the procedures will carry out their required tasks.
4. How data will be stored by the system, including the variables that will be used and the file structures required.

After an initial design has been created, this design should be tested and improvements made based on feedback from stakeholders and users.

Link

A number of techniques can be used to produce designs, some of which are described in the section on Flowcharts in this unit, and the use of standard symbol conventions and Structured English (pseudocode), is described in Learning aim B of this unit.

Implementation

Once the design for the software is completed, the task of coding the program can then begin. This is commonly referred to as the implementation stage. In a large software development project, this may involve a number of software developers (programmers), each of whom is responsible for creating the code for a different aspect of the software program.

The programmers will use a coding language in which to develop the software solution. There are several programming languages available to use, each with their own strengths and weaknesses. For example, Visual Basic is known as an event-driven programming language and is well suited to creating a graphical user interface (GUI) quickly within a Windows environment. However, a programming language such as C++ is a more advanced object-oriented language. Object-oriented programming (OOP) takes a different approach to the structure of a complex problem. OOP was developed in response to the difficulties that were experienced in creating highly complex systems. Most professional computer games engines such as Unreal Engine, CryEngine and Unity® use derivatives of the C programming family.

When developing a programming solution, the programmers will use the same programming language across the piece of software in order to create a consistent and coherent system. In this unit, you should use either a C-derived programming language or Python® to develop your software.

Links

See the section on 'Programming paradigms', for more about how a software developer will implement a programming solution to create software.

For more about programming, see *Unit 1: Principles of Computer Science*, 'Programming paradigms'.

For more about programming languages, see *Unit 1: Principles of Computer Science*, 'Types of programming and mark-up languages'.

For more about the choice of programming languages for computer games, see *Unit 14: Computer Games Development*, 'Software options and their effect on the development of computer games'.

Testing

As each part of the software program is created, it needs to be tested. This is the testing stage of the software development life cycle. Testing involves checking that all the functions and features of the software program work correctly.

Testing is important to check the quality of the finished software product and to ensure that the program does not contain any **bugs**. It is also important to ensure that what has been produced matches the organisation's and the users' requirements as outlined in the analysis stage.

Key term

Bug – a fault or error in a program which causes it to crash (end unexpectedly) or produce unexpected results.

Link

For more on how to test a software program, see the section on 'Test data', in 'Standard methods and techniques to develop designed solutions', and the section on 'Evaluation of software testing', in 'Evaluating a software development project'.

Discussion

It is not uncommon for testing to be overlooked or not completed as it should be. However, testing is crucial to ensuring that the program works as expected. In small groups, discuss your experiences of software programs unexpectedly 'crashing' or of errors occurring, and think about why this happened in each case. Did the error occur at the same point every time? Were you able to fix the problem? If so, how?

Evaluation

When the software program has been created and is in use, the process is not yet over. Even with careful testing, it would be very unusual for a program not to experience problems when it is first released. Therefore an evaluation stage needs to be carried out. Problems experienced by users need to be corrected and there may also be other improvements or additions to the program that may be required or desirable. For example, suppose you purchase a new car. It will require regular maintenance and servicing, and you may want to make additions to it such as better security features. It is similar with a software program. Once it has been released, the program still needs to be evaluated to check that it is working properly.

It will also need to be maintained to ensure that it continues to work and it will need to be updated to account for changes to related systems, or upgraded to meet new user requirements. This is why the evaluation stage is also sometimes called the maintenance stage. The fact that the software needs to be maintained implies that the life cycle continues, because updates or upgrades will need to go through the complete software development life cycle.

Link

See the section on 'Evaluating a software development project', for more on how the development process is evaluated once a software program has been created.

Determination of scope and size

Scope and size is about understanding the requirements of the software program and capturing all the necessary information that will be used as part of the development of a software program. Determining the scope and size of a software program is very important and should be done during the conception stage of the software development life cycle. By understanding the scope and size, a decision can be made as to whether it is beneficial and viable for a particular software development project to go ahead or not.

In order to understand the scope of a software project, fundamental questions must be asked: 'What will the software system do?' and 'What will it not do?'. These may sound like obvious questions, but they are important ones to consider. In effect, this is a matter of deciding which features are to be included in the software and what is to be left out.

Computers are very powerful machines and there are many features and facilities that you could include in a piece of software you are developing if you had endless time and an infinite amount of money. However, your resources are likely to be limited in terms of money, people and equipment, and also time-constrained. Given unlimited resources and money, you could create a massive piece of software that did everything, but this is unlikely, so you need to consider the realistic size and scope of the software.

Therefore, it is important to decide on the most important features. You should work out, through discussion with stakeholders and users, what the essential and desirable features are in order for the software to fulfil its main purpose. Once you have established a list of priorities within the organisation and the user requirements, it will be possible to create the software with the essential features as a first version. You would then put the remaining features on the 'wish list' for future versions, depending on time, money and other resources. This is how the scope and size of a software system is determined.

How to apply each stage of development to get the best results

It is essential to apply each stage of the software development process correctly in order to get the best results. Consider, for a moment, what would happen if this did not take place. For example, suppose you have analysed the current system, received all the requirements for the software product and then designed a solution. After this, you would go on to implement or code the solution. However, what if the customer did not like your design? What if you failed to identify an important requirement that was not elicited during the analysis stage? The consequences of this would mean that development would be pushed back, and the process would need to be repeated until it was correct. Carrying out further development work would take additional time and cost more money. Even if you were then able to produce a software solution that does fulfil all the organisation's and the users' requirements, the client may still be dissatisfied because it was late being released.

Therefore the crucial question we must ask ourselves is: 'How do we apply each stage to get the best results?'

The answer to this is communication, and the best method to employ to yield the best results is to use an iterative approach. This means that the software developers will follow a development process whereby an application is developed in small sections called iterations. Each iteration is reviewed and critiqued by the software team, the client and potential users. The evaluation of each iteration is then used to determine the next step in development.

The advantage of using iterative development is that the client and users are involved in the development process at each stage. Instead of waiting until the software application is a final product, when it may not be possible to make changes easily, problems can be identified and solved at each iteration and stage of development. This means that the problems are highlighted and resolved sooner, and are less likely to become an issue.

Research

Compare the agile software development process with the classic waterfall method. What do you consider to be the advantages and disadvantages of each development process?

PAUSE POINT Can you explain what the learning aim was about? What elements did you find easiest?

Hint Close the book and make a list of all the elements of the software development life cycle. Can you recall what steps are involved within each stage?

Extend How does the software development life cycle compare with different life cycle models? Which one do you feel is the best and why?

B Standard methods and techniques to develop designed solutions

Planning the design for a software program is a very important part of the software development process. The first step in planning the design is to determine what the main software program tasks will be, that is, what the software will do. At this stage of the process, it is also important to give some thought to how information is going to be stored, managed and processed.

Over the years, a number of different methods and techniques have been developed to model the way in which a software program works. Most of these methods involve creating diagrams, often starting with simple ones, and then building up to more complex diagrams as the understanding of how the system should work and what features are needed develops.

Flow charts and use of standard symbol conventions

This section will cover selecting and using software tools for developing flow charts to describe a system or solution. A flow chart is a diagram that represents a particular process in a system or software program. Software developers use flow charts to find solutions to problems, and they are often used in troubleshooting contexts. Producing flow charts is a very important activity in the software design process. A flow chart contains a mixture of symbols, which are all connected together to illustrate how data flows throughout a process.

It is important to note that there are universal standards (conventions) for how symbols are applied in flow charts. For example, it is universally recognised that a diamond shape represents a decision. It is this standardisation of symbols that makes flow charts a powerful design tool, as there is a common understanding of what all the symbols represent. Consequently, all software developers can interpret each other's flow charts easily, and they can therefore agree on approaches to solving problems, resolving ambiguities and making improvements to their software designs.

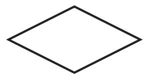
Use of British Computer Society (BCS) symbols

As explained above, a flow chart is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted by a shape and these steps are connected by lines and directional arrows. The symbols used in computing for flow charts are an industry standard adopted by the British Computer Society (BCS). This means that the same symbols are used by all computing professionals, which gives them a standardised approach to developing flow charts and facilitates a common understanding. Any software developer can view a flow chart that uses these symbols and logically follow the process from beginning to end.

Link

For more about the BCS, which is The Chartered Institute of IT, visit **http://www.bcs.org/**.

Table 4.1: BCS flow chart symbols and their uses

Symbol and use	Appearance
Processes: Normal processing steps, such as doing a calculation, are contained in a rectangle with a brief description of the step written inside. The rectangle has one arrow entering (from the previous step) and one leaving (going to the next step).	
Decisions: Where a choice or decision needs to be made (a section building block), a diamond shape is used, containing a question that describes the choice. One arrow enters the diamond shape and two leave it: one for the route taken if the answer to the question is 'yes' the other for if it is 'no.'	
Inputs: Processing steps which involve some input are shown in a parallelogram. They also have a brief description of the step written in the box. One arrow enters the parallelogram and one leaves it.	
Outputs: Output steps are shown as a box which is meant to look like a torn-off piece of paper.	
Connectors: All symbols (shapes) are linked by connectors, which are represented by arrows. Arrows are used to show the relationship between representative symbols.	
Start/end: A flow chart begins with a lozenge containing the word 'Start', with a single arrow leaving it. A flow chart ends with a lozenge containing the word 'Stop', with a single arrow entering it.	

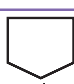
Flow charts should not contain any programming languages; only natural languages should be used (for example English). A limited use of general symbols is however acceptable in flow charts.

Figure 4.2 is an example of a flow chart that checks password input. In this example, the user has three opportunities to type the password correctly before the program identifies them as an unauthorised user.

Reflect

What is the very first symbol you have to write down before developing your flow chart?.

Tip

When using Microsoft® Visio  can be used as an off page connector, if your flowchart becomes too large to fit onto one page.

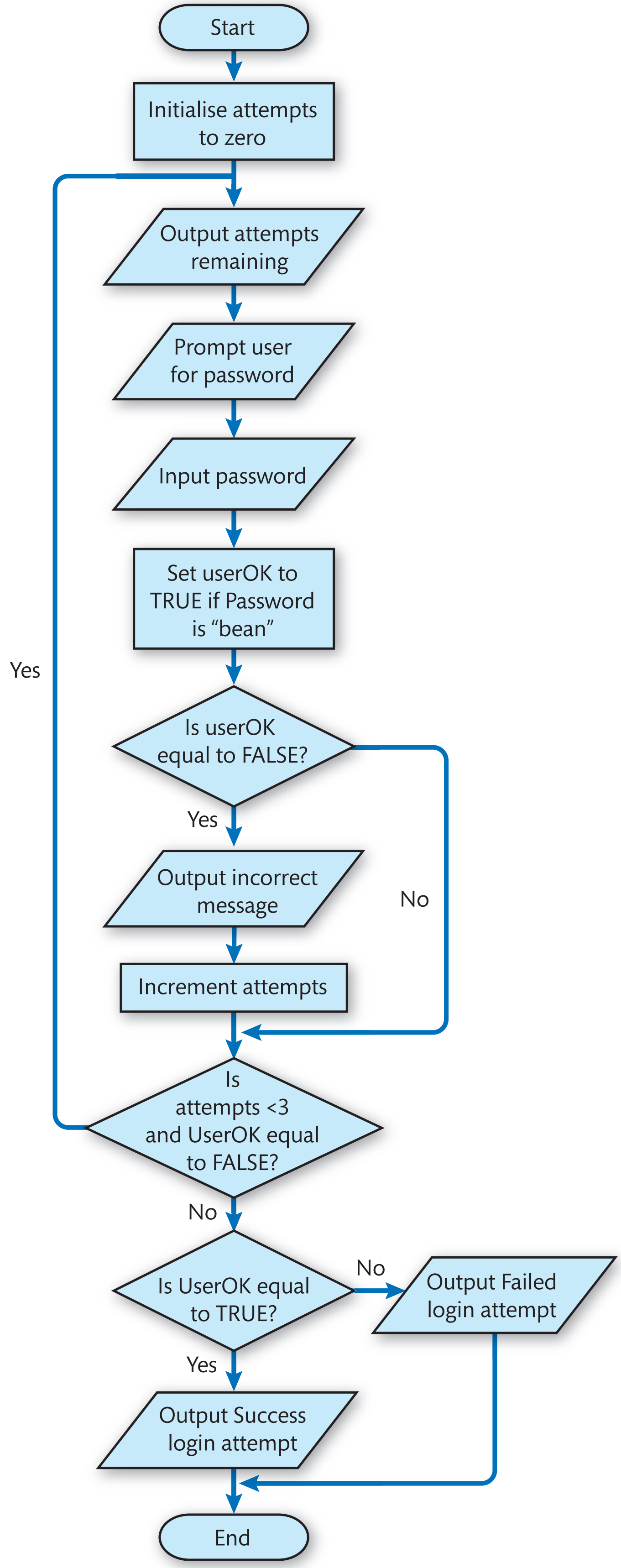


Figure 4.2: Flow chart process for password login

Assessment practice 4.1

A01 A02

Produce a flowchart for the following scenario using the standard BCS flow chart symbols.

Scenario

You work for a local supermarket and a customer wants to buy some groceries. You begin by greeting the customer and then scan the items that they wish to purchase. Once the items have been scanned, you inform the customer of the total.

Upon informing the customer of the total, you ask whether they would like to pay by cash, credit/debit card or contactless payment. If they decide to pay by cash, then you accept the payment and provide them with a receipt. However, if they decide to pay by credit/debit card, you begin by getting the customer to put their card in the chip and pin machine. If the card is declined, then the customer will have to pay by cash. If the card is accepted, then you provide them with a receipt. Similarly, contactless payment follows the same process for payment as paying by credit/debit card.

Once a receipt has been provided, you ask the customer if they require help with bagging their shopping. If they accept this offer of help, you contact a helper to bag their shopping. Alternatively, if they do not require any help, then they bag it themselves and you finish by wishing them a good day.

Plan

- What is the task? What am I being asked to do?
- Are there any symbols of which I am unsure? If so, make sure I research each symbol fully.

Do

- I know what it is that I am doing and what I want to achieve.
- I can adopt an incremental approach in which I start from the beginning of the scenario and work through to the end.

Review

- I can explain each stage in the development of the flow chart.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

Structured English (pseudocode)

Flow charts are useful, but one downside with them is that they rarely resemble the actual code that needs to be written. Pseudocode can be used to make the design resemble more closely the actual code that will be needed.

Pseudocode provides a design technique that is very similar to the code that will eventually be written. The name pseudocode comes from 'pseudo' (which means 'like' or 'a form of') and 'code.' It is an informal version of programming code that uses the structure of a programming language (for example decisions and loops), but does not worry about the strict **syntax** (rules) of the language. As it is written in English and is done so in a structured way (although not following the strict syntax of a programming language), pseudocode is also referred to as 'structured English'. (Flow charts can be thought as diagrammatical pseudocode.)

Figure 4.3 shows an example of pseudocode for a program that adds two numbers together and displays the result. Figure 4.4 represents the flow chart for the same program.

Key term

Syntax – a set of rules that is unique to each programming language which defines how commands and instructions should be structured and ordered.

```
PROGRAM CALCULATOR
BEGIN
     INPUT number A
     STORE number A
     INPUT number B
     STORE number B
CALCULATE total A + B = C
PRINT C
END
```

Figure 4.3: Pseudocode for a program that adds two numbers together and displays the result

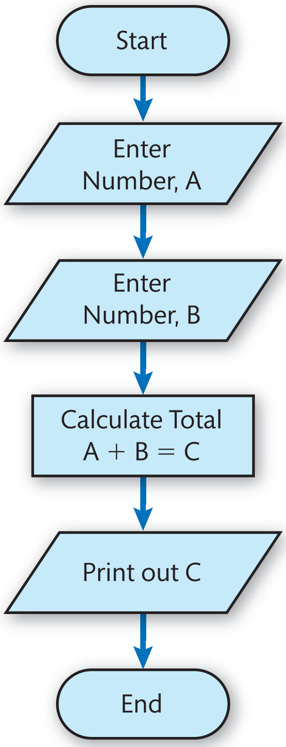


Figure 4.4: Flow chart for the program shown in Figure 4.3

Discussion

In small groups, consider which method you feel is best for developing a design for a computer program such as a basic calculator. As a software developer, would you consider the best method to be a flow chart or pseudocode? Justify your reasoning in your groups and present your findings to the class.

Writing pseudocode

It is important to understand that there is no set standard for writing pseudocode, and the writing of it is subjective. This is different to the use of flow charts, because flow charts have set conventions, that is, standards for how they should be developed. For example, if a team of ten software developers were designing a program for a calculator and all wrote their designs in pseudocode, it is highly likely that the results would be ten different pieces of pseudocode, even though they would all be for the same program.

One of the most important things a software developer should do when writing pseudocode is to ensure that their code has clarity. This is one of the primary goals of pseudocode, and one of the ways in which a software developer can achieve this is to use standard programming conventions. As you develop your pseudocode into actual code, you will need to transcribe it into a programming language, so the use of standard programming conventions to help structure your outline pseudocode will give it clarity.

Sequence

In pseudocode, a sequence means that one action is performed after another. Actions must be performed in a particular order so that the program will function correctly. The actions are performed from top to bottom in the sequence (order) in which they are written. When writing pseudocode, it is important to always follow the same programming conventions, for example indenting (see below) the code. Figure 4.5 shows how a sequence is applied in pseudocode.

```
AREA OF A RECTANGLE
BEGIN
     INPUT height of rectangle
     INPUT width of rectangle
     CALCULATE area as height multiplied by width
END
```

Actions are performed in sequence – one after another

Figure 4.5: A sequence in pseudocode

Key terms

Hierarchy – to sequence actions in the correct order.

Indentation – starting text further from the margin than the main body of the text.

Structure

The structure of pseudocode is very important. It must be written in such a way that makes it intuitive to follow, and thereby makes the code more readable. Thus, when writing pseudocode, you must ensure that the structure of it demonstrates a **hierarchy** and **indentation**.

Hierarchy

Hierarchy means to sequence things in order of importance. Setting a hierarchy in pseudocode enables you to represent the levels of importance or the proper sequence of events/actions. To understand this, see the pseudocode in Figure 4.6 in which the hierarchy of the program is incorrect as the levels of sequence have not been implemented correctly. The result is that it is harder for a programmer to follow the sequence, which could result in errors in the program.

```
CALCULATE GRADE
BEGIN
     IF grade greater than or equal to 70
        OUTPUT Distinction
     END IF
INPUT grade
END
```

Hierarchy of the program is incorrect because INPUT grade should appear before the if statement

Figure 4.6: Hierarchy in pseudocode

Indentation

Indentation is how the pseudocode is offset from the margin. By indenting the pseudocode, you make it more readable and intuitive to follow. This also helps

improve the hierarchy, as you can determine whether the actions are set in the right order because the code is clearer. If you failed to indent any code, whether it be actual code or pseudocode, it would be hard to read and maintain. See the example of indentation of pseudocode in Figure 4.7. Figure 4.8 shows the same code when it has not been indented.

```
CALCULATE STUDENT GRADES
BEGIN
     IF grade greater than or equal to 70
          OUTPUT Distinction
     ELSEIF grade between 60 and 69
          OUTPUT Merit
     ELSEIF grade between 40 and 59
          OUTPUT Pass
     ELSE
          OUTPUT REFERRAL
     END IF

END
```

Indenting the pseudocode makes it easier to read, and results in more maintainable code

Figure 4.7: Indentation in pseudocode

```
CALCULATE STUDENT GRADES
BEGIN
IF grade greater than or equal to 70
OUTPUT Distinction
ELSEIF grade between 60 and 69
OUTPUT Merit
ELSEIF grade between 40 and 59
OUTPUT Pass
ELSE
OUTPUT REFERRAL
END IF
END
```

By failing to indent the code it makes it much more difficult to read, making it difficult for the programmer to code the actual program

Figure 4.8: When the code in Figure 4.7 is not indented

Operations

Operations are pseudocode statements that you can add to a program to describe basic computing tasks in order to solve computing problems. Table 4.2 demonstrates the use of such operations.

Table 4.2: Operations in pseudocode

Pseudocode operation	**Explanation**	**Example**

BEGIN	When developing pseudocode to design a program, BEGIN is one of the first statements written. It denotes when the program has started.	`BEGIN`
		`  INPUT height of rectangle`
		`  INPUT width of rectangle`
		`  CALCULATE area as height multiplied by width`
		`END`

END	END is used to denote when the program will end.	`BEGIN`
		`  INPUT height of rectangle`
		`  INPUT width of rectangle`
		`  OUTPUT area as height multiplied by width`
		`END`

INPUT	INPUT is used to denote when a user has to type, or otherwise has to input, something into a program.	`BEGIN`
		`  INPUT num1`
		`  INPUT num2`
		`  INPUT num3`
		`  AVERAGE = (num1 + num2 + num3) / 3`
		`  OUTPUT AVERAGE`
		`END`

OUTPUT	OUTPUT is used by programmers to output or display information from the screen to the user.	`BEGIN`
		`  INPUT num1`
		`  INPUT num2`
		`  INPUT num3`
		`  AVERAGE = (num1 + num2 + num3) / 3`
		`  OUTPUT AVERAGE`
		`END`

PRINT	OUTPUT and PRINT can be used interchangeably by programmers. They mean the same thing and perform the same operation. It usually depends on a person's preference as to which one they decide to use.	`BEGIN`
		`  IF student's grade is greater than or equal to 60`
		`    PRINT "Passed"`
		`  ELSE`
		`    PRINT "Failed"`
		`  END IF`
		`END`

READ	READ and WRITE are used in **file handling**. READ is used where you wish to inspect a file or folder.
	In the example, the pseudocode reads in a record from a file and assigns it to a variable (called Record here).
	Each READ statement reads a record from the file.
		`BEGIN`
		`  READ MyFile.doc Record`
		`END`

WRITE	WRITE is used when you wish to add a file.
	In the example, a WRITE statement writes a record to the file.
		`BEGIN`
		`  WRITE MyFile.doc Answer1, Answer2, 'abc123'`
		`END`

Key term

File handling – programming to access files and folders from the hard drive of a computer or other storage.

Decisions

Decision making is a critical concept when it comes to software development and computer programming. Decisions are used in situations in which you are given two or more options and you need to select an option based upon given conditions.

Table 4.3: Decisions in pseudocode

Pseudocode decisions	Explanation	Example

IF	The IF statement allows programmers to introduce a decision, making it into a software program. This is sometimes referred to as selection.	`BEGIN`
		`  IF Answer = 10 THEN`
		`    SET Score TO Score + 1`
		`  END IF`
		`END`

THEN	THEN informs the programmer that, if a condition has been set, then something needs to happen. This is always the last word that appears in the line where the IF statement occurs.	`BEGIN`
		`  IF Answer = 10 THEN`
		`    SET Score TO Score + 1`
		`  END IF`
		`END`

ELSE	The ELSE statement is the final condition when all other conditions have been met. Therefore, this is the final line of an IF statement, though it may be omitted.	`BEGIN`
		`  IF login equal to “Kelvin” and pw = “12345” THEN`
		`    OUTPUT “Logged in”`
		`  ELSE`
		`    OUTPUT “Accessed denied”`
		`  END IF`

ELSEIF(ELIF)	An ELSEIF is a nested (subordinate) IF statement when working on multiple conditions.	`BEGIN`
		`  IF age is greater than 68 THEN`
		`    PRINT “Retired”`
		`  ELSEIF age greater than 17 THEN`
		`    PRINT “Adult of working age”`
		`  ELSE`
		`    PRINT “Child”`
		`  END IF`
		`END`

WHEN	Some programming languages use WHEN. It is used in the same way as an IF statement.	`BEGIN`
		`  WHEN Score = 5 THEN`
		`    End program`
		`  END WHEN`
		`END`

Repetition

- In computer programming, repetition (a loop) is where the program asks a question and then, if the answer requires an action, it is performed and the original question is asked again until the answer shows that the action is no longer required. For example, suppose that a program is written to check that all members of a gym are over the age of 18. This question is repeated continually and the action is performed in a loop until there are no more gym members’ ages to be checked.

Table 4.4: Repetition in pseudocode

<table>
<tr><th>Pseudocode repetition</th><th>Explanation</th><th>Example</th></tr>
<tr><td>FOR</td><td>FOR is a loop (iteration) where a process is repeated a specific number of times, which is determined by the program or the user. The loop 'counts' the number of times the process will be executed.</td><td><pre>BEGIN
  FOR Index FROM 1 TO 10 DO
  SEND ArrayNumbers[Index] TO OUTPUT
  END FOR
END</pre></td></tr>
<tr><td>REPEAT UNTIL</td><td>REPEAT UNTIL is a post-conditioned loop. It executes a command until the condition is true. The loop must execute at least once.</td><td><pre>BEGIN
  REPEAT
  SET Go TO Go + 1
  UNTIL Go = 10
END</pre></td></tr>
<tr><td>WHILE</td><td>A WHILE loop must repeat until a certain condition is met. If the condition is false at the beginning of the loop, the loop is never executed. This is useful for validating numeric entry.</td><td><pre>BEGIN
  WHILE Flag = 0 DO
  SEND 'All well' TO DISPLAY
  END WHILE
END</pre></td></tr>
<tr><td>WHILE NOT</td><td>WHILE NOT loops are used in Python® programming. They are used when trying to get a clean input from a user. This is useful when validating data entry. The program will keep trying until the user provides the correct input.</td><td><pre>BEGIN
  INPUT age
  WHILE NOT age is a digit
    PRINT "Number is invalid"
    INPUT age
  END WHILE
  PRINT "Number is valid"
END</pre></td></tr>
</table>

Developing pseudocode

Pseudocode must be developed in order for it to be efficient and effective. A programmer will develop pseudocode to solve a problem, and then a team of programmers will utilise this pseudocode to develop a solution to the problem. What would happen if the pseudocode design was incorrect or not efficient? It would have serious implications later on in the development process. Therefore, you should pause and consider how your pseudocode might be developed further, to make it more effective and efficient.

Improving the effectiveness and efficiency of pseudocode

In order to improve the effectiveness and efficiency of pseudocode, it is essential, during the analysis stage of the software development process, that all the requirements and functionality of the program are understood. If this has not been done properly, then the pseudocode that is developed will either be incorrect, have key elements missing (i.e. it will not be effective) or might not carry out actions in the most efficient way for the intended purpose. This is why it is crucial that the client and users are heavily involved during the analysis stage.

Once all the organisation's and the users' requirements have been ascertained, design tools such as pseudocode can then be used. In order for the code to be as effective and efficient as possible, it is best to tick off all requirements of the program in the pseudocode. Once this has been done, it is not uncommon for the pseudocode to be shared with others in the development team for review. Sometimes, in order to improve the effectiveness of the pseudocode, **comments** are used to explain why a solution has been developed in the way it has.

Key term

Comments – explanations of the pseudocode or code. In programming terms, commenting is ignored by the computer but allows a programmer to know what the code will do.

Quite often, alterations are made to pseudocode by other software programmers. This is usually done in small teams where the pseudocode is discussed and explained. At this point, alterations can be made to refine the code. For example, if a program must accept some kind of data entry, such as a calculator, what would happen if a person typed in text rather than numbers? There must be some kind of validation embedded within the pseudocode to prevent the program from crashing. This could be omitted by one programmer, but it is likely to be picked up by others.

Link

For more about validation, see the section on 'Validating data', in 'Programming paradigms'.

Identifying and fixing errors within pseudocode

Identifying and fixing errors in pseudocode is crucial before the software program is written. Consider what would happen if there was not enough proper vetting of the pseudocode; what would the likely effect be on the program you are developing? The answer is that you would have a program containing errors, which can make the development process more time consuming and expensive, because these errors would need to be fixed at a late stage in development. Therefore, before coding can take place, the pseudocode designs must be checked to ensure that there are no errors.

When developing pseudocode (or virtually anything for that matter), it is hard to be critical over what you have created yourself. This is why, in software development, it is extremely unlikely that you will be working by yourself in large-scale programming projects. Instead, you will be working as part of a team and responsible for an element of the computer program. The programmers in a project team will often meet to discuss their designs and the best way to proceed. They will also review each other's work to identify and fix errors in the pseudocode, before the actual coding development takes place.

Research

Research what software design techniques there are, other than flowcharts and pseudocode. What are the advantages and disadvantages of each technique?

Test data

In this section, you will learn how to select appropriate tests and test data to produce test plans for software solutions.

Software testing is the process of checking that all the features and functions of a program work as they should do and give the correct results to the tests they are subjected to. What is meant by terms such as 'work as they should' and 'correct results' needs to come from the original program specification that was agreed by the client/users and the software developers before the program was written.

Making sure that a software program works properly may sound like a fairly simple task but, except for the very simplest of programs, software testing is a complex and involved task which requires planning.

To test a program, you first need to produce some test data that you can use to test the program. Choosing test data involves choosing some input values and then manually working out the expected output that the program should produce with these chosen inputs. The program should then be run using these input values and the expected outputs should be compared with the actual ones that the program produces. If there is a difference between the expected outputs and the actual outputs, then the program has failed the test and will need to be modified so that the expected and actual outputs match.

What to test

There are many testing processes that a software developer can use to test a software program. It is not feasible to use all the different forms of testing to test a program, as it would take far too much time. Instead, software developers will select a form of testing that is best suited to the problem which they have to solve (the purpose of the software). Developers may test all these aspects to different extents, depending on the time they have available and the key requirements of the software.

Functionality testing

Functionality testing is a testing process used within software development in which a program is tested to ensure that it conforms to all its user functionality requirements. It does this by using the requirements specification that was written in the analysis stage of the software development life cycle. The functionality of a software program is the actions and calculations that it should be able to carry out. Another phrase for functionality testing is '**black box testing**' which, just like functionality testing, ensures that all the requirements are met.

Key term

Black box testing – a testing method (also known as functionality testing) which tests all the functions of the program without the need to peer inside the internal structure of the code.

Stability testing

Stability testing is often referred to as endurance testing. This method of testing checks to see if the software program can continuously function well for an acceptable period of time or longer. The program will be run continuously to ensure that the program is robust and will continue to work as expected under prolonged conditions.

PAUSE POINT Write out the pseudocode needed to calculate an individual's monthly loan repayment. The user will need to type in: the loan amount; how many months they are borrowing for, and the monthly interest rate.

Hint To determine the loan repayment formula you must apply the following formula:

$$P = \frac{r * A}{1 - (1 + r)^{-N}}$$

Where:

P = Payment Amount

A = Loan Amount

r = Monthly Rate of Interest

N = Number of Payments

Extend Does your pseudocode validate against incorrect data entry? For example, what would happen if you typed in incorrect data such as characters, which the formulae can't calculate? See if you can add some validation within your pseudocode to prevent the program from erroring.

Usability testing

Usability testing is testing of the software program by different users. Quite often, this will involve the client and users (employees and/or customers). They will be asked to check that the software works as intended and that they do not have any issues when using the program. This is essentially an evaluation of the software and is a good opportunity to refine and improve the program, as necessary. Usability testing (or user testing as it may be called) is invaluable because it gives developers direct feedback on how real users deal with the system, that is, information about how well it works in general and any specific issues that the test users encountered.

Choice of test data

For all the forms of testing outlined above, tests must be formulated that will establish whether the software performs its required functions correctly: that is, whether it is stable and usable. Test data will be needed in order to carry out these tests.

Before performing a test, you need to decide what data you are going to include in your test cases. It is not normally possible to perform tests with every single possible piece of data. Instead, the software developers will choose from a range of data. There are three main categories of test data: normal, abnormal and extreme. Developers need to know how the program will respond to all three types of data.

Normal test data

Normal test data is data that you would normally expect as an input value for your program. For example, if your program were for a gym, people would need to input their ages on the application form for the gym and normal test data would include values such as:

- 21
- 40
- 55
- 68.

Abnormal test data

Abnormal test data are input values which are incorrect entries for that part of the program. For example, when inputting ages of gym members, the following would all be incorrect because a number that could be a person's age is required:

- 31/01/1984 – because it is a date rather than an age.
- 250 – because, although it is a number, it is not realistic for a person's age.
- Forty five – because it is a text value where a numerical value is expected. (Using numbers consistently would allow mathematical analysis to be carried out using the data set of gym members' ages.)

If a date of birth were called for instead of an age, then 31/01/1984 from the list above would be acceptable (normal test data) but 31/02/1984 would be abnormal, because there are only 28 (or 29) dates in February.

Extreme test data

Extreme test data is data at the boundary between typical (normal) data and invalid (abnormal) data. For example:

- a textbox where someone's name is to be entered: extreme values might include 'Tc', '123456789', 'xxxxx', or a very long name.
- a validation rule where only numbers between 10 and 100 can be entered: typical test data on the boundaries would be input values of 100, 99, 10 and 11.

Expected test results from a range of testing methods

In order to compare the expected test results against the actual test results from differing testing methods, they must be recorded somehow. Quite simply, this is done using a document called a test plan. A test plan is created before the testing begins and lists the test data that will be input and the results that are expected when the program carries out specified functions. An example template for a test plan is shown in Table 4.5.

Once the test plan has been created, the test data listed in it are used as input values to test the program. The results of each test are recorded on the test plan. If the expected output and the actual output match, then the program has passed that test. However, if they differ, then the program has not passed the test and the error messages produced must be entered in the plan.

Link

For more about validation, see the section on Validating data, in Programming paradigms.

Tip

When developing your test plan, it is always recommended that you have your page in a landscape orientation. This gives you more space to complete your tests, because you can see more columns at once.

The completed test plan is then passed back to the programmer to investigate the failed tests and correct the problems with the software. The corrective actions that the programmer takes should be noted in the last column of the test plan.

Once the program has been corrected, the program should be tested again. You might imagine that when the program has been corrected and is to be tested again, only the tests that it failed the first time need to be done again. However, is possible that the programmer has inadvertently introduced further faults while making corrections. So, to be sure that the program works properly, the complete set of tests should be done again. This process should be repeated until the program passes all the tests in the test plan.

Discussion

In small groups, consider why it is a bad idea for a program to be tested by the person/people who created it.

Table 4.5: Test plan template

Test number	Purpose of test	Test data	Expected result	Actual result	Comments

Assessment practice 4.2

A02

Develop a test plan (with a minimum of 20 tests) to test the functionality of the monthly repayment calculator that you completed for the Pause Point at the end of the section on Structured English (pseudocode). This should include a range of tests from normal, abnormal and extreme test data. Try to develop an equal spread of tests using normal, abnormal and extreme test data. This means that you will have a consistent testing approach throughout your test plan.

Develop a series of questions, both quantitative and qualitative, that you could ask a user to test a program against. This form of testing is usability testing and is useful for gauging other people's opinions and their perceptions of the program that you have developed.

Plan
- What is the task? What am I being asked to do?
- Have I included a range of test data?

Do
- I know what it is that I am doing and what I want to achieve.
- Make sure that I include a minimum of 20 tests and that I include a range of tests.

Review
- I can justify why I have selected each test.
- I can explain why testing is required.

C Software design considerations

Computer users expect software programs to be of a high standard. Therefore, it is important that software programmers think carefully about the design and the techniques they use to develop the software. Conversely, it is equally important that programmers are aware of poor design features and the implications of embedding poorly designed solutions into programs. This section looks at the ways in which you can measure the quality of the software you design by looking at design concepts, as well as the impact of embedding poorly designed solutions.

Design concepts

There are fundamental design concepts which (where possible) should be applied to all software programs. Conversely, there are concepts indicative of poor design which should be avoided. As a software developer, you need to be aware of these poor design concepts so that you know not to employ them in your designs. In this section, you will explore both good and poor design concepts.

Selecting and applying common good practice design concepts

The following good practice design concepts are what software developers must consider and employ in their designs prior to the implementation stage of the software development life cycle. These key design concepts play an important role in ensuring that the end product is one that is fit for purpose and meets the requirements of the client. The common good practice design concepts are detailed below.

Compatibility

Programmers must think about how software programs will run on different types of computer systems, as well as how they interact with other software. Software can react and perform differently, depending on the type of computer system it is running on. Software is considered highly compatible if it can be executed (run) on any type of computer system.

Extensibility

Once a software program has been developed for a client, it is quite normal for clients to want additional functionality added to their software program. For example, if you consider an electronic point of sale (EPOS) system, then new sale items, new categories of items or other functionality may need to be added. Therefore, it is important to design any software program so that it remains possible to add new capabilities without the need to write a brand new program or make major changes to the underlying architecture of the existing program. This ability to add new capabilities to a program at a later date is the program's extensibility.

Efficiency

When you design a software program, a key feature will be how well it performs, that is, how efficient it is, in terms of speed, processor time, memory space and interaction with storage media.

The overall performance of a software program (that is, how efficient it is, and how fast it carries out its instructions) will be a measure of the quality of the software. The way in which the code is written has a big influence on this. For example, a software program that has efficient and well-structured code will perform better than one with poorly structured code. Usually, poorly structured code will have more lines of code than is necessary to arrive at the same outputs: that is, it is less efficient.

Informative

It is crucial to ensure that any programs you develop are informative to the end user; that is, tell the user what they need to know. If a computer program was not informative, it would make it hard for a user to know what precisely to input into the program. In other words, it will not be intuitive to the needs of the user. For example, consider a program created to calculate someone's body mass index (BMI). If the program did not inform the user of what type of weight measurement they should use (that is, kilograms rather than pounds), then it would not be informative. This kind of information needs to be specified within the program to inform the user of what kind of input value they need to enter.

Fault recovery

Fault recovery is the ability of a software program to continue operating in the event of a failure of some of its components. When developing a software program, developers should try to ensure that the program can continue its intended operations, but at a reduced level, rather than failing completely, if some kind of fault or error occurs. This can be achieved by anticipating conditions in which the software program may encounter problems, and thereby developing measures to enable it to cope with these conditions. For example, in a Windows®-based operating system, when some kind of error occurs,

Windows® can sometimes reboot itself in safe mode. This is a troubleshooting option for Windows® that starts your computer in a limited state.

High maintainability

A well-written software program can easily be modified in order to carry out necessary alterations to the code. Such alterations may be made in order to correct, refine or adapt the existing code. That a software program can successfully be modified in these ways shows it to have high maintainability.

It is important for a software developer to write a program that is easy for them or someone else to change in the future. For example, comments should be added to the code and meaningful names should be used as variables.

Link

For more about variables, see the section on 'Managing variables', in 'Handling data in a program'.

Modularity

When developing software, it is good practice to write code that is well defined, and to write independent components which can be implemented in isolation before being integrated to form the complete software system. This practice ultimately improves the maintainability of the program. This design concept is the modularity of a program. You are likely to encounter this practice if you work in the computing industry. As a software developer on a project team, you could be responsible for one component part of the system (a module). This practice allows for the division of work in a software development project.

Reliability

People depend on software programs to provide them with accurate information because often the data they supply is used to inform important decisions. For example, sales figures can determine whether or not someone receives a bonus. Therefore the accuracy and reliability of a software program's output is another crucial measure of quality. Reliability of a program is achieved by thorough testing to ensure that program behaves in the way in which it was intended.

Reusability

Within computer programming, reusability is a key quality measure that helps a programmer to use existing coding assets. Reusability of program code saves time and resources and reduces **redundancy** by taking advantage of coding that has already been created. **Object-oriented programming** makes use of reusability, and is therefore one of the main types of programming used in large-scale development projects.

Key terms

Redundancy – code within a computer program which is not necessary.

Object-oriented programming – a programming methodology based on the concept of 'objects', which are structures that contain data.

Robustness

Software needs to be able to cope with processing large amounts of data reliably. It must also be able to handle unexpected input values in a controlled way. Having these characteristics indicates that a program is robust. An example of robustness is a software program responding correctly to incorrect data entry. When developing software, it must always be validated to ensure that the program responds correctly to unexpected data entry. If you were developing a calculator program, but mistakenly typed in some text by accident, it is important that the program responds appropriately to this incorrect data entry by informing the user that what they typed in was incorrect.

Link

For more about validation, see the section on 'Validating data', in 'Programming paradigms'.

Correctness

A key principle of any software program is to respond to the wishes of a user and to give the appropriate output. It is possible for a program to work correctly, but still output incorrect information. For example, consider the formula for determining the area of a shape. What would happen if the formula in your program was incorrect, but the code compiled and worked? When designing a software solution, it is therefore also important to ensure that correct output will be derived from correct user input.

Usability

It is important that a software program is easy to use for its intended purpose: this is the usability of the program. Software should be designed carefully to guard against problems such as a poor layout of the user interface, a lack of instructions to help the user or error messages which do not explain how a problem can be fixed. Users should be involved in testing the software for usability as part of the design process.

Research

In pairs or small groups, compare and contrast procedural programming, event-driven programming and object-oriented programming. Which do you believe is the best programming method and why?

The features, impact and implication of embedding poorly designed solutions

In the previous section, common good practice design concepts were identified, which software programmers must consider and try to incorporate into their design prior to the implementation stage of the software development life cycle. It is equally important that they also take into consideration poor design concepts. Understanding these poor design concepts and how they occur will enable a software developer to know how to avoid these pitfalls in their own programming.

Rigidity

The term rigidity is derived from the word 'rigid', which means to be inflexible. When related to software development, rigidity means that if a developer wished to make changes to the program, each change would cause an unexpected result to occur somewhere else within the program. Therefore a simple change to a program, which might take only a couple of days to amend in a flexible program, could grow into a week-long marathon of changes as software developers try to resolve all the unexpected changes due to rigidity.

In order to avoid creating a 'rigid' program, software developers tend to opt for an object-oriented approach to programming. This type of programming involves breaking down elements of a problem into classes. These classes collect data and functions, which are like templates for creating objects. One of the principle advantages of object-orientated programming is that it enables programmers to create new objects that inherit many of their features from existing objects. This makes object-oriented programs more flexible and less rigid. It is easy to make changes to the program because it derives its features from existing objects in the program which have already been tested and implemented.

Fragility

Fragility is a term closely related to rigidity. Fragility is the tendency of the software program to error or break in many different places any time a change is made to the program: that is, it is fragile and breaks easily. As a consequence, if you are using a fragile piece of software in your daily work, your managers will always be worried that it will break. This makes maintaining the software exceptionally difficult, and any fix that is implemented can ultimately introduce more problems than it actually solves.

In order to avoid producing a 'fragile' program, software developers can employ a combination of various validation techniques to ensure that any unexpected result does not cause the system to crash. For example, if you are creating a menu system from 1–6 and someone types in '8', then the program should be able to deal with this unexpected result, and prompt the user to try again. This is achieved through validation.

Link

For more about validation, see the section on 'Validating data', in 'Programming paradigms'.

Immobility

The concept of immobility within software development describes the inability of the program to reuse software from the same or different programs. A software developer may decide that they would like to use some functionality that another software developer has written either for another component of the same software system or for another that the company is developing. However, the extract that they require may have too much baggage that it depends on. For example, there could be a program that compiles and works perfectly well, but in which the programmer has failed to use any appropriate naming conventions for variables or functions. Therefore, this makes it hard for someone else to understand what each part of the code (and therefore what the program) does. This is because the software developer has to try to separate the desirable parts of the code from the undesirable parts, and it ultimately takes more time than simply writing new code. Therefore, the software has to be rewritten instead of being reused, which is a waste of time and money.

Inconsistency

Inconsistency is a concept used throughout software development, and it is important to avoid the pitfalls of an inconsistent program. Inconsistency in a software program is where the parts of a program do not follow the standard conventions of what has previously been mapped out. For example, if a program utilised the same colour, font and font size throughout, but then, in the next part to be designed, used a completely different style, it would result in an inconsistent program. The end result is a software program that is not intuitive to the needs of the user because users will expect the design and functionality to be consistent (which also makes it easier to use).

Dissatisfied users

The whole premise of writing a software development program is to meet the needs of the users of the program, as specified by the client. If you, as a software developer, fail to achieve this, then it will result in dissatisfied clients and users. Therefore, when developing software programs, you must employ as many good practice design concepts as possible and avoid the poor ones. For example, you should ensure that all designs and layouts are consistent. If you adopt good practice design concepts and avoid the pitfalls of poor design, then this will ultimately lead to satisfied clients and users.

Code readability

Often programmers will work on each other's code. This is quite normal when you work in a team to develop software for a software development company. Therefore you need to write your code in such a way that it can be read and understood by others. Well-written code can be understood by others and reused in other programs. Moreover, you will need be able to understand your own code if you come back to it months or even years later. There are several techniques that are available to programmers to help create readable code.

The features, applications, impact and implications of adopting code readability when developing code

When developing code, it is essential that you adopt and use appropriate coding conventions, no matter which programming language you decide to use. By adopting these good coding conventions, you will make the program more readable and easier for another programmer to understand. These good coding conventions are standards that are inherent to all programming languages. Good programmers always abide by these standards, because the consequences of not abiding by them would mean that their programs would be hard to understand, develop further and maintain. The main four good coding conventions are detailed below.

Maintainability

For a software developer, maintainability has several different characteristics. Code is maintainable if it is understandable, intuitive and adaptable.

Understandable

In order for code to be understandable, another programmer must be able to pick up the code to read and easily discover its purpose and general approach without a walkthrough by the original programmer. The best way of achieving this is to comment on your code. You will find out more about commenting later in this section.

Intuitive

For code to be intuitive, it just needs to make sense and flow in a logical manor. A good example of this is how a modular approach can be used to separate code into subsections which can be used or referred to later. This makes the code easy to follow and the whole program more maintainable.

Adaptable

Adaptable code is where a program can easily respond to any changes which may be required. For example, in the pause point task where you had to create a monthly loan calculator, what if the repayments had to be made quarterly instead of monthly? Would the program adapt easily to this need? By using proper coding conventions, we can ensure that the program can adapt to changes efficiently and quickly, without a whole rewrite of the code.

Naming conventions

When developing code for a software program, it is important to use the correct naming conventions throughout. For example, if you were to create a variable for storing someone's age and called that **variable** 'num1' then this would not make sense to anyone else who tried to read your program. Moreover, if the developer of the program came back to the program a week later, they themselves may find it hard to remember and identify what 'num1' really represents. Therefore it is important, when creating any names within a program, such as variables or functions, that they are given proper names. For example, if 'num1' was used to store someone's age then 'age' would be a more appropriate name than 'num1'. Figure 4.9 shows how naming conventions are used in code.

Key term

Variables – are used to store information in a computer's memory, which allows a programmer to retain, change and access this information as the program runs.

Link

For more about variables see, the section on 'Managing variables', in 'Handling data in a program'.

For more about functions, see the section on 'Function calls', in 'Control structures'.

```
int age = 18;
float pi = 3.1416f;
double moreaccuratepi = 3.14159265359;
bool lighton = false;
```

Figure 4.9: Naming conventions in code

Research

In pairs or small groups, research what 'Hungarian Notation' means in the context of naming conventions. What are the advantages and disadvantages of using this approach to naming?

Indentation

Indentation of code is where code is 'pushed in' (indented from the margin) to make it much more understandable and easier to read. If you do not indent any code, it would be extremely hard for a programmer to read the code. As you can see from Figure 4.10, the indentation of the code has been highlighted. It shows how the IF statements and brackets are clearly lined up with one another so that you can see how the bits of code relate to one another.

```
#include <iostream>

int main()
{
    //switch case statement
    int switchnumber = 0;
    std::cout << "Please select one of the following numbers: 1, 2 or 3" << std::endl;
    std::cin >> switchnumber;
    switch (switchnumber)
    {
    case 1:
        std::cout << "You selected the number 1" << std::endl;
        break;
    case 2:
        std::cout << "You selected the number 2" << std::endl;
        break;
    case 3:
        std::cout << "You selected the number 3" << std::endl;
        break;
    default:
        std::cout << "You did not select a listed number" << std::endl;
        break;
    }
}

```

Figure 4.10: Indented code

Figure 4.11 shows the same code as in Figure 4.10 but it has not been indented. As you can see, this code is harder to read.

```
#include <iostream>

int main()
{
//switch case statement
int switchnumber = 0;
std::cout << "Please select one of the following numbers: 1, 2 or 3" << std::endl;
std::cin >> switchnumber;
switch (switchnumber)
{
case 1:
std::cout << "You selected the number 1" << std::endl;
break;
case 2:
std::cout << "You selected the number 2" << std::endl;
break;
case 3:
std::cout << "You selected the number 3" << std::endl;
break;
default:
std::cout << "You did not select a listed number" << std::endl;
break;
}
}
```

Figure 4.11: Code from Figure 4.11 without indentation

Commenting / code annotation

Whenever a software developer writes some code, they should comment on the code as they write it, as a matter of good practice, because the comments explain what the code is doing and make it easier to follow. Therefore, if someone else were to read the code, they would be able to understand what the developer was trying to accomplish. Figure 4.12 shows some C++ code which has been commented on. As you can see, the commenting appears in a different colour from the rest of the code. This enables the **compiler** that checks the code to ignore the comments, but they are there so that programmers understand the purpose of the code.

Key term

Compiler – a compiler is a special program that processes statements written in a particular programming language and turns them into machine language or machine 'code' which computers use to run programs.

```
//implement libraries
#include <iostream>
#include <fstream>

//minimises code later in program
using std::cin;
using std::cout;
using std::endl;
using std::ofstream;
using std::ifstream;

//function and procedure declaration
//parameter passing by reference utilised to keep program modular
void openMainMenu(double &balance);
double getBalance(double &balance);
void exitProgram(double &balance);
double withdrawMoney(double &balance, bool &overdraft);
double depositMoney(double &balance);
bool requestOverdraft(bool &overdraft);

int main()                      //Begin program
{
    double balance = 0.00;      //Initialising the balance to 0
    getBalance(balance);
    openMainMenu(balance);
    return 0;
}
```

Figure 4.12: Code with comments

Discussion

In pairs or small groups, discuss what other methods you know of for commenting on code. For example in C++, commenting is represented by two forward slashes '//'. Also consider what makes a compiler different from an interpreter.

Factors that contribute to the quality of code

There are many different factors that contribute to the overall quality of code. As a software developer it is important that you understand what these factors are and, where possible, try to ensure that you adopt good practices in relation to them. The key factors that contribute to the quality of code are the following.

Efficiency

Efficiency is directly related to the performance and speed of running the software. By developing more efficient code you can improve the performance and running speed of a piece of software. No one likes to use software that takes too long to perform an action. Code is made more efficient by removing unnecessary or redundant lines of code, by developing more reusable code and by using appropriate data types, functions and looping in appropriate places.

Readability

Readability of code means ensuring that the coding that you develop is intuitive and easy to understand. This can be accomplished by a variety of methods such as commenting your code, using correct naming conventions and indenting the code. All these methods play an important role in making sure that the code is readable, which, in turn, contributes to the overall quality of code.

Key term

Infinite loop – a loop is a sequence of instructions that is continually repeated until a certain condition is reached. There is no exit from an infinite loop so it repeats indefinitely.

Robustness

By making code robust, you can ensure that the program can stand up to scrutiny and not error at any point. It can be very frustrating when you create a program and it crashes when you input some data, so good quality code will be robust and will not crash or error. Moreover, if you create a program that becomes stuck in an **infinite loop**, it is not robust because it does not function well. You must ensure that the code that you develop is fully validated and can deal with any unexpected data entry.

Usability

Usability is a program's potential to accomplish the goals of the user; specifically, how easy it is to use. Therefore, when developing software code, you must make it easy for users to achieve their purpose in using the software. For example, if a program is to be used to calculate the volume of a cuboid, you must make it clear to the user what information they need to input and why. If you simply asked the user to type in a number without explaining what that number is for, it would be hard for them to ascertain what is happening in the program. Therefore, in order to make code usable, you must make it clear within the code what it is intended to do (that is, what its purpose is) by stating clearly what information is required by the user and why.

Portability

Portability is how easy it is to take the software from one environment and transfer it to another. Well-written code should be capable of functioning in different environments. To best achieve portability, it is advisable to compile the program on at least two different platforms, early and often during development. Typical choices of environment are Visual C++ on Windows® and on Linux™. As compilers are not shared between the different environments (operating systems), you will detect non-portable code and have the opportunity to make it portable before the completion of the program.

Assessment practice 4.3

A02

Write a small software program in three different programming languages. These should be:

- C++ (or any C-derived programming language)
- Python®
- C#.

Your job is to ensure that you apply the concepts of code readability to these three programs. The program can be a simple entry-based system which will add two numbers together.

When developing your program, try to consider and use at least three good practice design concepts and apply them to your program. Moreover, be sure to include the features of code reusability in your programs, e.g. using appropriate naming conventions.

Use online resources to help you, for example YouTube and Code Academy, the latter of which has a wealth of coding tutorials in different programming languages to support you in learning these languages.

Plan

- What is the task? What am I being asked to do?
- Have I got the programming languages to do this? If not, there are online emulators which can do this.
- Research good design concepts and apply at least three to my program.

Do

- I know what it is that I am doing and what I want to achieve.
- I will create a program in which I have applied good design concepts.
- I can apply the conventions of code readability to my program.

Review

- I can explain how code readability improves a program.
- I can explain how the program is improved by using good design concepts.

D Programming paradigms

In this section, you will learn how to build and develop accurate, efficient and effective computer code to meet client requirements and solve problems. You must use standard coding conventions in either Python® 3.4 (or subsequent later versions) or C family programming languages to build and develop your software solutions. The coding examples in this unit are developed in C++.

Handling data within a program

When you are developing computer programs (no matter which programming language you use), it is expected that, at some point, you will need to handle data. You will need to handle data using a number of different methods which are discussed in this section.

Defining and declaring constants and variables

When programming, it is important to know how variables and constants are defined and declared because computer programs use them to store data within a program. Data can also be variable or constant within programs and functions.

- Variables are something that may change in value and are used to store information. For example, if you were making a computer game, you could make a variable called 'score.' This variable would be used to store the information about the number of points you have been awarded while playing the game.

- Constant values always remain the same, that is, they are unchanging. For example, if you were writing a program and you knew that the value would not change, you would use a constant to represent this. A good example of this may be VAT (20%), as VAT will be the same value throughout the program.

Alphanumeric strings

A string is a sequence of characters that can include text or numbers, and hence it can be called an alphanumeric string. C++ enables you to use powerful tools that allow a user to manipulate these strings. However, you must include a standard **library** facility for the alphanumeric string to be used. Therefore, whenever you want to use alphanumeric strings or string manipulation tools, you must provide this in the #include directive. #include is a command that essentially pastes previously written code into your program. It must be followed by an appropriate library. For example, if we want to include the printf() function we must use the <stdio.h> library. Figure 4.13 shows an example of an alphanumeric string library using the #include directive.

```
#include <string>        //used for string related functions

int main()
```

Figure 4.13: Alphanumeric string library

Key terms

Library – a package of code that can be reused by programs many times over.

Data types – used to specify the kind of information that the programmer needs to store. Booleans, integers and strings are just some examples of data types.

Once you have an alphanumeric string library in place, you can use the string in a similar way to built-in **data types**. It can be used to assign, compare, reassign or perform other types of string manipulation.

Arrays

In programming, an array is a form of data structure, which means that it is a way of collecting together data items of the same type.

An array can be one-dimensional or multi-dimensional. Programmers use arrays to help solve complex problems because they provide an easy way to access similar data values using the same name in any order the programmer chooses.

Link

For more about data structures and arrays, see 'Data structures', in 'Programming paradigms'.

Boolean

A Boolean is a data type used to declare a variable whose value will be set as true or false. In order to declare such a value, you use the keyword bool. The variable can then be initialised with the starting value. Figure 4.14 shows an example of a Boolean in use.

[*] Boolean.cpp

```
#include <iostream>
using namespace std;
//--------------------------------------------------------------
int main()
{
    bool checkMachineIsOn = true;

    cout << "The Machine is working therefore, it is "
         << checkMachineIsOn << endl;

    bool checkMachineIsOff = false;

    cout << "The Machine is not working therefore, it is "
         << checkMachineIsOff << endl;
    return 0;
}
```

Figure 4.14: Boolean code

In this code, two variables are being declared, 'checkMachineIsOn' and 'checkMachineIsOff' and defined as Boolean data types. This means it can only return true and false: 'checkMachineIsOn' is assigned to true and 'checkMachineIsOff' is assigned to false. Figure 4.15 shows the output from the code in Figure 4.14.

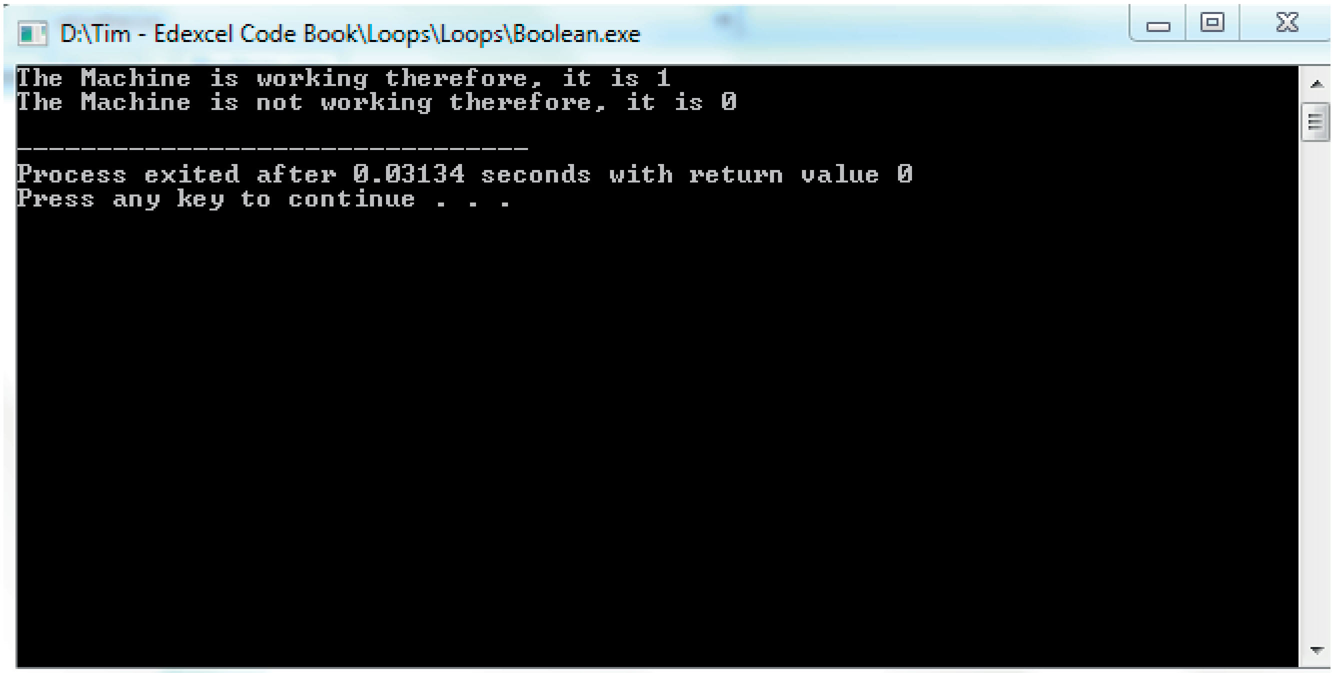


Figure 4.15: Boolean output

We have assigned 'checkMachineIsOn' to true, which the output shows as 1. Likewise we have assigned 'checkMachineIsOff' to false, which the output shows as 0. cout is used to output the variables.

Characters

C++ offers a predefined data type that is capable of a single byte – in other words exactly one character, such as 'a' or 'A.' You declare a character data type using the keyword char. Figure 4.16 shows a character defined by char.

```
#include <iostream>

int main()
{
    char yesorno = 'y';
}
```

Figure 4.16: Character defined by char

Date/time

C++ does not have a standard library for providing a date or time data type. Instead, C++ inherits the functions for date and time manipulation from its origin language C. To access date and time-related functions and structures, you need to include the header <ctime> in your C++ program. For example, Figure 4.17 demonstrates how to incorporate the date and time using the header <ctime>.

Research

Using what you learnt on how to create and output a Boolean variable, create a program yourself that makes use of a Boolean. For example, see if you can develop a username and password program, that, when the correct credentials are entered, will allow a user to log in. If the wrong credentials are entered, then the program will inform the user they have typed in their credentials incorrectly.

Link

Similar code for presenting the date / time in Python® appears in *Unit 1: Principles of Computer Science.*

Time.cpp

```
#include <iostream>
#include <ctime>

using namespace std;

int main( )
{
    // current date/time based on computer running program from
    time_t now = time(0);

    // convert now to string form
    char* datetime = ctime(&now);

    cout << "The date and time is: " << datetime << endl;
}
```

Figure 4.17: Date and time using <ctime>

Key term

Real number – a real number can be a whole number, a rational number (a fraction or a decimal) or an irrational number (one which cannot be expressed as a fraction or decimal, e.g. π or the square root of 5). Infinity and imaginary numbers (e.g. the square root of minus one) are not real numbers.

Floating point (real)

Floating point is a data type which can hold a **real number**, such as 3456.0, -4.345 or 0.03453113. It is referred to here as 'floating point (real)' because it holds real numbers. The 'floating' part of the name floating point refers to the fact that the decimal point can 'float' in terms of its position. Therefore it is a variable that can hold a number of digits before and after the decimal point.

In C++ programming there are three different floating point data types (see Table 4.6).

Table 4.6: Floating point data types

Category	Type	Typical size
Floating point	Float	4 bytes
	Double	8 bytes
	Long double	12 or 16 bytes

Figure 4.18 shows a floating point number declared in C++.

```
#include <iostream>

using namespace std;

int main()
{

    float pi = 3.1416f;
    double moreaccuratepi = 3.14159265359;

    cout<<"pi is equal to: "<<pi<<endl;
    cout<<"moreaccuratepi is equal to: "<<3.14159265359;

    return 0;
}
```

Figure 4.18: Floating point number in C++

In C++ when “int main” has been declared it must return a value. This is because the “int” part is known as a return type, so it expects the function to return an integer. Therefore, returning 0 ensures the function gets what it wanted, a returned integer.

Research

In pairs or small groups, research what the ‘f’ signifies in Figure 4.18.

Discussion

In small groups or in pairs, consider why you would use different floating point data types. In other words why would you not simply use long double for all floating point numbers?

Integers

When we require a program to represent whole numbers, we use the data type integer. The integer data type uses the keyword **int**. The code in Figure 4.19 is an example of how to declare an integer data type.

```
#include <iostream>

using namespace std;

int main()
{

    int age = 18;
    cout<<"Age is assigned "<<age;

    return 0;
}
```

Figure 4.19: Integer data type

Two basic rules need to be adhered to when writing integer values in programs.

1. Decimal points cannot be used. For example, even though 35 and 35.0 have the same value, ‘35.0’ is not of the data type int because it includes a decimal point.
2. Commas cannot be used in integers. For example 34,098 must be written as ‘34098 when using the int data type.

Objects

Objects are integral to object-oriented programming (OOP). When programming using OOP, the programmer has to examine the problem in a less traditional way. Traditional procedural programming is when a programmer writes a step-by-step series of instructions (i.e. a procedure) which they want their program to carry out. In OOP on the other hand, a programmer seeks to break down elements of the problem into a number of **classes**. In procedural programming, data and functions are usually kept separate. The OOP approach does things a little differently by packaging the data and its function together (this is called encapsulation). The collection of data and functions in OOP is called a class, and essentially acts like a template for creating objects.

Objects are created through a process called instantiation. This sounds complex, but, in reality, instantiation simply means making solid instances (objects) from a class. Figure 4.20 is an example of classes and objects in use, where a class (‘cuboid’ in this case) is declared and the object is accessed using the direct access operator (.).

Key term

Class – a term used in object-oriented programming whereby data and functions are collected together to form a class.

```
#include <iostream>

using namespace std;
class cuboid
{
    public:
        double length;   // Length of a cuboid
        double width;  // Width of a cuboid
        double height;   // Height of a cuboid
};

int main()
{
    cuboid Cuboid1;         // Declare Cuboid1 of type Cuboid
    cuboid Cuboid2;         // Declare Cuboid2 of type Cuboid
    double volume = 0.0;      // Store the volume of a Cuboid here

    // Cuboid1 specification
    Cuboid1.height = 7.0;
    Cuboid1.length = 3.0;
    Cuboid1.width = 14.0;

    // Cuboid2 specification
    Cuboid2.height = 11.0;
    Cuboid2.length = 17.0;
    Cuboid2.width = 20.0;

    // volume of Cuboid1
    volume = Cuboid1.height * Cuboid1.length * Cuboid1.width;
    cout << "Volume of Cuboid1 : " << volume <<endl;

    // volume of Cuboid2
    volume = Cuboid2.height * Cuboid2.length * Cuboid2.width;
    cout << "Volume of Cuboid2 : " << volume <<endl;
    return 0;
}
```

Figure 4.20: Object created from the class 'cuboid'

Research

Create a triangle class which stores the height and base. Develop this to create two instances of 'triangle'.

Records

In programming, arrays allow a programmer to define variables that combine several data items of the same kind. Alternatively, within C-derived programming, you can use a 'struct', which is another user-defined data type that allows a programmer to combine data items of different kinds.

Structs are used to represent a record. For example, assume you wish to keep track of students in a school. You might want to track the following attributes about each student:

- studentID
- surname
- forename
- age.

In order to access any member of a struct, we use the member access operator (.). The access operator is coded as a full stop between the struct variable name and the struct member that we wish to access. Figure 4.21 shows a struct that uses information about students in a school, as in the example above.

Link

For more about data structures and records, see 'Data structures', in 'Programming paradigms'.

```
#include <iostream>
#include <cstring>

using namespace std;
struct Student
{
    char studentID[10];
    char forename[30];
    char surname[30];
    int age;
};
int main( )
{
    struct Student Student1;        // Declare Student1 of type Student
    struct Student Student2;        // Declare Student2 of type Student
    // Student1 Record
    strcpy(Student1.studentID, "AND12345");
    strcpy(Student1.forename, "Kelvin");
    strcpy(Student1.surname, "Andrews");
    Student1.age = 25;

    // Student2 Record
    strcpy(Student2.studentID, "BEA23456");
    strcpy(Student2.forename, "Mick");
    strcpy(Student2.surname, "Kumar");
    Student2.age = 19;

    // Print Student1 Details
    cout << "Student1 ID : " << Student1.studentID <<endl;
    cout << "Forename : " << Student1.forename <<endl;
    cout << "Surname : " << Student1.surname <<endl;
    cout << "Age : " << Student1.age <<endl;

    // Print Student2 Details
    cout << "Student2 ID : " << Student2.studentID <<endl;
    cout << "Forename : " << Student2.forename <<endl;
    cout << "Surname : " << Student2.surname <<endl;
    cout << "Age : " << Student2.age <<endl;
    return 0;
}
```

Figure 4.21: Record data type.

In the example we have used a struct called, 'Student' and we can create new members of 'Student' using the member access operator '.' The reason programmers use a struct is because, if we were, to use variables to store the information we would need to declare additional variables for every person we add. This means the program would have so many variables it would get out of control and difficult to maintain. A struct allows a programmer to group variables of mixed data types together into a single unit.

Sets

Sets are containers that allow a programmer to store unique elements following a specific order. To be able to use sets, you need to include the library #include<set>. The example in Figure 4.22 demonstrates the use of a set which lists the numerical elements in order, and removes repeated elements.

Link

For more about string handling, see 'String handling functions', in the section on 'Built-in functions', in 'Programming paradigms'.

Strings

Strings are sequences of characters that can include text or numbers. In C++ you can manipulate strings by using functions. For example, if you wanted to copy one string to another you would use the **strcat** string function.

```
#include <iostream>
#include <set>
using namespace std;

int main()
{
    set<string> demoSet;
    cout<<"Demonstration of a set"<<endl;

    demoSet.insert("1");
    demoSet.insert("2");
    demoSet.insert("2");
    demoSet.insert("3");
    demoSet.insert("5");
    demoSet.insert("6");
    demoSet.insert("3");

    cout<< "Set contains:";
    while (!demoSet.empty()) {
        cout<< ' ' <<*demoSet.begin();
        demoSet.erase(demoSet.begin());
    }
    return 0;
}
```

Figure 4.22: A set in use

Managing variables

In this section, you will look at how to manage variables within code. You will consider local and global variables as well as naming conventions.

Local and global variables

In programming, a variable is used to store and retrieve data from the computer's RAM. Every variable should have a unique (and meaningful) name. In order to reserve enough RAM for the variable, you must select an appropriate data type.

For example, if you want to store someone's age in a variable you would type int age;. This line of code is called a declaration. It essentially reserves enough RAM to store an integer (whole number) and lets you refer to that reserved RAM by the name that you have picked: age.

Local and global variables define how 'visible' the variable is. In a large program split into a number of different modules, a global variable would be visible to all modules, whereas a local variable would only be visible within the module it was declared in.

Professional programmers prefer to use local variables where possible because any faults with the variable will be in one particular module. This isolation of the variables makes a program easier to debug.

Naming conventions

A naming convention is a set of rules for choosing the names of variables or functions. They must be given meaningful names. There are very good reasons why you must name anything within a program appropriately. One important reason why naming conventions are used is to reduce the effort needed to read and understand source code: that is, to improve code readability.

For example consider again int age;. Common sense tells you that this is storing an integer for a program and, specifically, that this will be someone's age. Now consider the following example of code: int num1;. This would be permissible. However, what is num1? It is impossible to ascertain what num1 is, whereas in the previous line of code (int age;) you know that it is storing information about someone's age. Therefore a naming convention int age; should be used rather than int num1; to make the code meaningful and therefore more readable.

Arithmetic operations

In all programming languages there are arithmetic operations (mathematical expressions) that are used to perform number calculations. It is unlikely that the arithmetic operations will differ from one language to another. Nevertheless, you need still be aware of how they must be applied to any particular programming language that you are using. In this section, you will look at how to select, apply, use and interpret arithmetic operations in computing structures to process data.

Mathematical operators

The arithmetic operators that can be used in C++ are detailed in Table 4.7. The following step by step will show you how to create a software program for a calculator using these arithmetic operators.

Table 4.7: Arithmetic operators in C++

Arithmetic operator	C++ implementation
Add	+
Subtract	-
Divide	/
Multiply	*
Percentage	%

Step by step: Creating a calculator program using arithmetic operations

6 Steps

1 Open C++ and type the following code to include your standard library and **namespace**.

```
# include <iostream>
using namespace std;
```

2 To start the program type the following.

```
int main()
{
```

3 Type the following to declare your variables.

```
char op;
float num1, num2;
```

4 Type how the program will accept input and display output.

```
cout << "Enter operator either + or - or * or /: "<<endl;
    cin >> op;
    cout << "Enter number1"<<endl;
    cin >> num1;
    cout << "Enter number2"<<endl;
    cin >> num2;
```

5 Type the following case statement to determine which arithmetic operation to perform.

```
switch(op) {
        case '+':
            cout << num1+num2;
            break;
        case '-':
            cout << num1-num2;
            break;
        case '*':
            cout << num1*num2;
            break;
        case '/':
            cout << num1/num2;
            break;
        default:
            /* If operator is other than +, -, * or /, error message is shown */
            cout << "Error! operator is not correct";
            break;

```

6 Type the following to end the program. Then compile and run your program.

```
}
    return 0;
}
```

Research

Try developing the code above to use if statements, as opposed to a case statement. Which method do you think is best and why?

Key term

Namespace – the line of code using namespace std; is useful to declare at the start of a C++ program, because it means you can write just cout instead of std::cout. This will ultimately save you time and take up less code.

Relational operators

In most languages, relational operators are a set of operators that make direct comparisons between values. The result of this type of comparison can only be true or false. Table 4.8 shows a list of these relational operators and their C++ implementation.

Table 4.8: Relational operations in C++

Relational operator	C++ implementation
Equal to (test for equality)	==
Not equal to (to test for inequality)	!=
Greater than	>
Less than	<
Greater than or equal to	>=
Less than or equal to	<=

Tip

In some programming languages, 'not equal to' is implemented as <> rather than!=.

Boolean operators

Boolean operators provide you with the capability to test multiple conditions. For example, if you were to create a login screen, you would need to test for two conditions: whether the username is equal to something AND whether the password is equal to something. Table 4.9 lists these logical operators and their C++ implementation.

Table 4.9: Logical operators in C++

Relational operator	C++ implementation
NOT	!
AND	&&
OR	||

Date and time

In C++ programming, there is no library for a date or time data type. Instead, C++ utilises the <ctime> header library combined with additional code to determine the actual date and time. It also uses arithmetic operations to perform this.

Link

For more about date and time implementation, see Date/time in Defining and declaring constants and variables in the section on Handling data in a program, in Programming paradigms.

Built-in functions

There are many built-in functions within programming languages. C++ has many built-in functions to help programmers add functionality to their programs.

Arithmetic functions

To assist programmers, C++ has built-in arithmetic functions, which are used to perform mathematical operations. You will now look at some examples of these.

Random function

The random function uses the built-in function rand() to generate a random number. The random number is derived from the <cstdlib> built-in library, as shown in Figure 4.23.

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main ()
{
    int randomNumber;

    randomNumber=rand();
    cout << "Output a random number " << randomNumber;

    return 0;
}
```

Figure 4.23: Random function

Range function within a range

The programmer can also specify a random number within a given range. The example cited in Figure 4.24 uses the built-in function rand() and adds on %15+1; . This will enable the program to randomly generate a number between 1 and 15.

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main ()
{
    int randomNumber;

    randomNumber=rand()%15+1;      // Randomizing the number between 1-15.
    cout << "Output a random number between 1-15: " << randomNumber;

    return 0;
}
```

Figure 4.24: Range function over a range

Round function

The round function uses the setprecision() function, and is derived from the <iomanip> library. The example in Figure 4.25 rounds π to 3 significant figures. If we changed the '3' in the program to '1' it would simply output 3.

```
#include <iostream>
#include <iomanip>      //used for precision function

using namespace std;
int main()
{
        double precisionnumber = 3.14159;
        cout << std::setprecision(3) << precisionnumber;
}
```

Figure 4.25: Round function

Truncation function

The truncation function uses the trunc() function which is derived from the <cmath> built-in library. The example in Figure 4.26 takes π expressed to 6 decimal places and removes the decimal part to give 3 (it is truncated). The function never rounds upwards, so trunc(5.999) would become 5.

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
        cout << trunc(3.141516) << std::endl;
return 0;

}
```

Figure 4.26: Truncation function

String handling functions

A string is a number of characters that are joined together. The string can be composed of any number of valid symbols. Some programming languages may define a limit for the length of the string. In C++ there is a built-in library called <string> and within this there are built-in functions that allow us to manipulate strings.

Concatenation and length

Concatenation means to take two or more separately located entities and place them next to each other, side-by-side, so that they can now be treated as one.

Length returns the number of bytes in the string. For example 'Testing 123' has a length of 11. (The actual size of the string may be greater once it is encoded and stored.)

The following step by step shows how to manipulate a string via string concatenation and string length.

Step by step: Creating string manipulation via string concatenation and string length

6 Steps

1 Open C++ on your computer and type the following code for your standard libraries.

```
#include <iostream>
#include <string>        //used for string related functions
```

2 Once you have completed this, type the following to start your program.

```
int main()
{
```

3 To demonstrate string concatenation, type the following below the code you have just entered in step 2.

```
    //string concatenation
    std::string firstword = "Hello";
    std::string secondword = " World!";
    std::cout << firstword + secondword;
```

4 To demonstrate string length, type the following below the code you have just entered in step 3.

```
    //string length
    std::string name = "Grace Hopper";
    std::cout << "The length of the string is: " << name.length();
    std::cout << "The letter at position 3 is: " << name.at(3); //at starts at 0, so 3 is   actually the 4th letter
```

5 To demonstrate string length, type the following below the code you have just entered in step 3.

```
    system("PAUSE");
    return 0;
}
```

6 Compile and run your program

Position function

A position function allows you to find the exact position of a character within a string. In the example in Figure 4.27, the program would return 'c' – not 'a', because the position function always starts at 0.

```
#include <iostream>
#include <string>        //used for string related functions

int main()
{
    //string length
    std::string name = "Grace Hopper";
    std::cout << "The letter at position 3 is: " << name.at(3);
}
```

Figure 4.27: Position

Figure 4.28 highlights all the elements within the word 'Grace'. The first letter of Grace is 'G'. The position of the first letter is defined as 0. 'c' is the fourth letter, so its position is defined as 3.

0	1	2	3	4
G	r	a	c	e

Figure 4.28: Position

String conversion

String conversion allows us to change a string into a number such as an integer or a float, and vice versa. Remember that a float is a data type that can hold a real number. Figures 4.29–4.32 show to carry out these string conversions.

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    //integer to string
    int integernumber = 999;
    std::string convertedstringinteger = std::to_string(integernumber);
    cout<<integernumber;
}
```

Figure 4.29: Integer to string

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    //float to string
    float floatnumber = 12.345f;
    std::string convertedstringfloat = std::to_string(floatnumber);

    //converted to integer
    cout<<floatnumber;
}
```

Figure 4.30: Float to string

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    //string to an integer

    string height = "1.80";
    int heightinteger = std::stoi(height);  //converted to integer
    cout<<height;
}
```

Figure 4.31: String to integer

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    //string to float

    string height = "1.80";
    float heightfloat = std::stof(height);  //converted to float
    cout<<height;
}
```

Figure 4.32: String to float

General functions

It is possible to open, input from or write to files using C++. For example, if you want to write something to a text document, you can do this using the <fstream> library, which allows a programmer to read from and write to files.

Open and write functions

The example in Figure 4.33 highlights how a C++ program can open a file and then write to it.

```
#include <iostream>
#include <fstream>                //Needed to read and write to files

using namespace std;

int main () {
  ofstream myfile;
  myfile.open ("test.txt");
  myfile << "Writing to a file";
  myfile.close();
  return 0;
}
```

Figure 4.33: Open and write to a file

Link

Similar coding for open, read, write, close appears in *Unit 1: Principles of Computer Science.*

Print function

In C++ there are lots of libraries to choose from, and one library that appears frequently is <iostream>, which is used for cin and cout operations. cin is used to take in information and cout is used to output or print information. The coding example in Figure 4.34 demonstrates this.

```
#include <iostream>

using namespace std;

int main( )
{
   char string[] = "Hello World";

   cout << "Value of string is : " << string << endl;
}
```

Figure 4.34: Print function

Link

Similar code for the print function in Python® appears in *Unit 1: Principles of Computer Science.*

Range function

In programming languages, a range is used to determine a list of numbers between set amounts. In Python® programming there is a dedicated range function that achieves this, range(), whereas C++ uses a rand() function to achieve this.

Link

For more about the range implementation in C++, see 'Range in Arithmetic function', in the section on 'Built-in functions', in 'Programming paradigms'.

Built-in library use to add functionality

A function is a group of statements that, when put together, perform a task. Just about any C++ program has at least one function which is **main()**. In C++ the standard library provides many built-in functions that a program can call. For example, the function strcat() can be used to concatenate two strings, while the function sqrt() is a library function to calculate the square root of a number. These built-in functions provide additional functionality which a programmer can use easily, meaning that the program can perform these operations without the programmer having to code them in manually.

Key term

main() – any C++ program must contain a function named main, which is the designated start of the program.

Validating data

In software programming, there are various validation techniques that can be used to ensure that data entered into a program is sensible and reasonable. You should note that validation of data does not ensure the accuracy of the data. To check the accuracy of data, it must be verified, which is different to validation.

Link

Further examples of Validating data are found in *Unit 1: Principles of Computer Science.*

Validation check techniques

In this section, you will look at programming techniques that a software developer can use to validate their programs. Validation check techniques are a way of enhancing your programming to ensure that the programs you create are robust and do not crash.

Data type

A data type is used to specify what kind of information you wish to store. For example, you can use integers to hold numbers or strings to hold text. The advantage of this is that the programming language will check that the data going into the variable is of the same type as you specified. For example, if you asked for someone's age (an integer) and they typed in their name (a string) the programming language would give you an error message.

Using data types and having to specify them can flag up an error quickly. This allows a programmer to identify the problem straight away and take immediate action. It is important to note that the actual names of data types will vary from language to language, but common categories are typically found (see Figure 4.35).

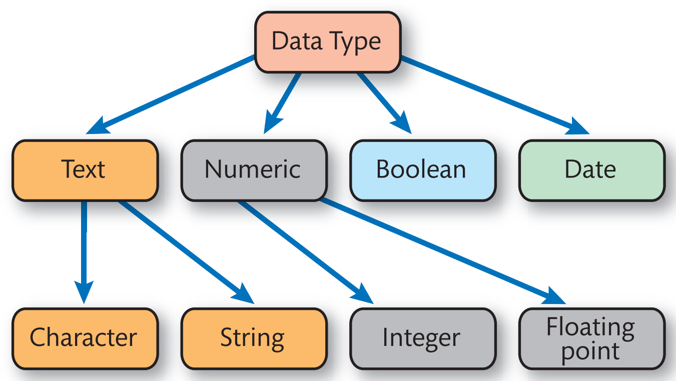


Figure 4.35: Data type categories

Range check

When programming, you may want to set a range or a limit to examine user input. You can set a minimum and maximum range to achieve this. For example, you may wish to set a range for people who can apply to a take out a credit card. The range check would ensure that only individuals between the ages of 18 and 100 can apply for a credit card.

Link

For more about the range function, see 'Range' in 'Arithmetic functions', in the section on 'Built-in functions', in 'Programming paradigms'.

Constant

In addition to variables, it is possible to create another type of identifier: a constant. As the name suggests, constants do not change their value once the program starts running.

For example, in Figure 4.36, VAT is declared as a constant and given the value of 20. A program using this declaration will use the constant VAT whenever you want to refer to the value 20.

```
#include <iostream>

using namespace std;

int main( )
{
   const int VAT = 20;

   cout << "VAT is equal to : " << VAT;
}
```

Figure 4.36: VAT as a constant

There are two advantages of this:

1 it improves the readability of the program code
2 if you want to change VAT, you only have to alter the constant's declaration, not find every occurrence of 20 in the code. This helps to improve the accuracy and validity of data within a program.

Boolean

The name of this validation technique recognises the field of mathematical logic developed by nineteenth century English mathematician George Boole. A Boolean value is either true or false (yes or no), reflecting the 0 and 1 binary values used by computer systems.

Post-check actions

Post-check actions (or conditioning) occur when a conditional statement is placed after the actions. The actions in a post-check conditioned loop will always work at least once. In C++ a post-check loop can be created by using a Do...While statement. Figure 4.37 shows an example of a post-check action.

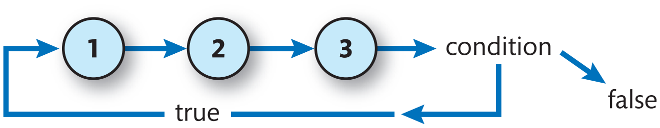


Figure 4.37: Post-check action

In the example in Figure 4.38, a post-check condition is used to repeat the loop while the counter is less than or equal to 10. This is sometimes referred to as REPEAT in programming (see the section on 'Control Structures', in 'Loops'). Each cycle of the loop outputs the counter's current value (starting from 0) and increments the counter. The loop stops when the condition is no longer true (when the counter reaches 11). The resulting screen output is shown in Figure 4.39.

A REPEAT...WHILE loop or DO...WHILE loop is similar to a WHILE loop with the exception that the test condition occurs at the end of the loop. This means that it guarantees that the body of the loop always executes at least once.

Link

For more about REPEATs, see the section on 'Loops', in 'Control structures'.

```
#include <iostream>

int main()
{
    int z = 0;
    do
    {
        std::cout << z << std::endl;
        z++;
    } while (z <=10);
}
```

Figure 4.38: Post-check condition used to repeat the loop while the counter is less than or equal to 10

```
0
1
2
3
4
5
6
7
8
9
10
```

Figure 4.39: Output from post-check in Figure 4.38 (once the loop has stopped because the condition is no longer true)

Research

Create a new program that makes use of a do while loop (post-check action) and adds two numbers together until a user enters 0.

Case statements

A case statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being 'switched on' is checked for each case. Case statements are similar to nested IF statements but they work in different ways. The use of nested IF...ELSE statements are common but can lead to unnecessary over-complication. Case (or switch) statements simplify things by making it possible to pick a single matching value from a list of possibilities.

Case statements are generally preferred to IF statements because they can make individual comparisons against each possible matching value. Moreover, when you present the same code using ELSE...IF, it looks much tidier and is less reliant on using brackets, which can make your code look overly complicated and cumbersome when you have a long nested IF statement. Figure 4.40 shows an example of a case statement.

Link

For more about IF, ELSE, ELSEIF, see 'Branches', in the section on 'Control structures', in 'Programming paradigm's.

```
switch (switchnumber)
{
case 1:
    std::cout << "You selected the number 1" << std::endl;
    break;
case 2:
    std::cout << "You selected the number 2" << std::endl;
    break;
case 3:
    std::cout << "You selected the number 3" << std::endl;
    break;
default:
    std::cout << "You did not select a listed number" << std::endl;
    break;
}
```

Figure 4.40: Case statement

```
if (ifnumber == 1)
{
    std::cout << "You selected the number 1" << std::endl;
}
else if (ifnumber == 2)
{
    std::cout << "You selected the number 2" << std::endl;
}
else if (ifnumber == 3)
{
    std::cout << "You selected the number 3" << std::endl;
}
else
{
    std::cout << "You did not select a listed number" << std::endl;
}
```

Figure 4.41: Nested IF statement

As you can see by comparing Figure 4.40 with Figure 4.41, a case statement appears less cumbersome and clearer than a nested IF statement.

PAUSE POINT With the aid of the flow chart in Figure 4.2 in the section on 'Flow charts and use of standard symbol conventions', use the concepts you have learnt to make this program.

Hint See if you can extend the program to accept multiple users logging in.

Extend Once you have got it working in your programming language of choice (e.g. C++), see if you can develop it in another language (e.g. Python®). The syntax will be very similar.

Control structures

Control structures are a way of analysing, developing and improving the effectiveness of code. A variety of programming techniques form these control structures, for example loops, branches (including IF statements) and functions.

Loops

In programming, a loop is a sequence of instructions that is continually repeated until a certain condition is reached. Sometimes, this is referred to as iteration. For example,

if you wanted to increment a number from one to ten, outputting each part would require the same code to be repeated ten times, and the only thing that would change is the output number (from one to ten). Loops allow you to repeat the process with each number until this reaches ten.

In programming there are three fundamental types of loop:

- WHILE loop
- REPEAT (DO...WHILE) loop
- FOR loop.

REPEAT loop

REPEAT is a form of a DO...WHILE loop. The difference between a WHILE loop and a DO...WHILE loop is that a WHILE loop checks a condition before it executes given statements of the code. A DO...WHILE loop differs because it allows a programmer to execute a loop body before checking a given condition.

Link

For more about DO...WHILE loops, see 'Post-check actions', in the section on 'Validating data', in 'Programming paradigms'.

WHILE loop

A WHILE loop is an example of a pre-check condition whereby the condition statement is placed before the actions (see Figure 4.42).

Link

For more about WHILE and DO...WHILE loops, see the section on 'Post-check actions,' in 'Validating data'.

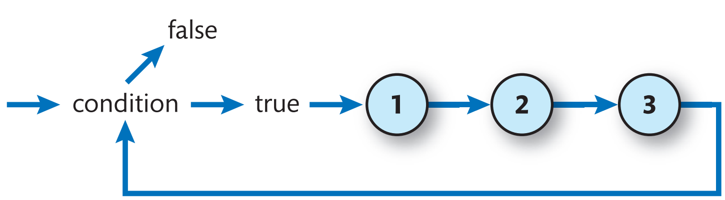


Figure 4.42: Pre-check condition

Placing the condition at the start of the loop has an interesting effect. If the loop condition is found to be false at the beginning, its actions will never be processed. The example in Figure 4.43 shows a WHILE loop incrementing a number from zero to ten.

```
#include <iostream>

int main()
{
    //while loop will count from 0 to 10
    int x = 0;
    while (x <= 10)
    {
        std::cout << x << std::endl;
        x++;
    }
}
```

Figure 4.43: WHILE loop

Research

Create a program that makes use of a While loop. Specifically, create a program to find the factorial of a positive integer entered by a user.

FOR loop

A FOR loop is a repetition control structure that allows you to write a loop that needs to execute a specific number of times. The syntax for this is detailed in Figure 4.44.

```
for ( init; condition; increment )
{
   statement(s);
}
```

Figure 4.44: Syntax for a FOR loop

The process associated with a FOR loop (as in Figure 4.44) is detailed below.

1. init is executed first (and only once). This allows you to declare and initialise any loop control variables.
2. The condition is evaluated next. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and the flow of control jumps to the next statement after the loop.
3. After the body of the loop executes, the flow of control jumps back to the increment statement. This statement allows you to update any loop control variables.
4. The condition is now evaluated again. If it is true, the loop executes and the process repeats itself (body of loop, then increment step and then, again, condition). After the condition becomes false, the FOR loop terminates.

Figure 4.45 shows the complete FOR loop.

```
#include <iostream>
using namespace std;

int main ()
{
    // for loop execution
    for( int x = 0; x < 11; x = x + 1 )
    {
        cout << "value of x: " << x << endl;
    }

    return 0;
}
```

Figure 4.45: FOR loop

Research

Create a program using a FOR loop that decrements from 20 to 2, and outputs only even numbers.

BREAK out of loop

A BREAK statement can be used to force a loop to terminate, even if the condition for its end is not reached. It can be used to end an infinite loop or to force a loop to end before its natural end. A BREAK is shown in Figure 4.46.

```
// break loop example
#include <iostream>
using namespace std;

int main ()
{
  for (int n=10; n>0; n--)
  {
    cout << n << ",";
    if (n==3)
    {
      cout << "countdown stopped!";
      break;
    }
  }
}
```

Figure 4.46: BREAK out of loop

Branches

The ability to control the flow of a program, that is, allow it to make decisions on what code to execute, is valuable to the programmer. An IF statement allows you to control whether a program enters a section of code or not, based on whether a given condition is true or false. This is why it is sometimes referred to as 'selection'.

A good example of how an IF statement is useful is when programming a login system. A user should only be allowed to enter a login once they have correctly inputted the username and password. If they do not, some output should inform the user to try again. All of this is achieved through an IF statement.

IF statements

The structure of an IF statement is shown in Figure 4.47.

```
if ( TRUE )
  Execute the next statement
```

Figure 4.47: Structure of an IF statement

THEN statements

In some programming languages, when you start with an IF and you check a condition, THEN is put at the end of the line to inform the program what is to happen next. This does not happen in C++ but is common in other languages such as Visual Basic.

ELSE statements

When the condition in an IF statement evaluates to false, it is often necessary to execute code different from the code

that would be executed when the statement evaluates to true. The ELSE statement specifies the code that should be executed if the IF statement is false. The structure of an ELSE statement is shown in Figure 4.48.

```
if ( TRUE ) {
   // Execute these statements if TRUE
}
else {
   // Execute these statements if FALSE
}
```

Figure 4.48: ELSE statement

ELSE...IF statement

Another use of an ELSE statement is when using an ELSE...IF statement. You would use an ELSE...IF statement when there are multiple conditional statements that may all evaluate to true, yet you want the body of only one IF statement to execute. You can use an ELSE...IF statement following an IF statement. If the first IF statement is true, the ELSE...IF will be ignored, but if the first statement is false, it will then check the condition for the ELSE...IF statement. If the IF statement was true, then the ELSE statement will not be checked. It is possible to use numerous ELSE...IF statements to ensure that only one block of code is executed. An example of an ELSE...IF statement is shown in Figure 4.49.

```
if ( <condition> ) {
   // Execute these statements if <condition> is TRUE
}
else if ( <another condition> ) {
   // Execute these statements if <another condition> is TRUE and <condition> is FALSE
}
```

Figure 4.49: ELSEIF statement

Step by step: Creating a grade determiner using IF statements in C++

5 Steps

1 Open C++ on your computer and, on the first line, type the standard library.

```
#include <iostream>
```

2 On the next line, type the following. (Remember: this will help you to not duplicate std when you want to use cout or cin.)

```
using namespace std;
```

3 Begin the program by typing

```
int main()
   {
```

4 Type the remaining code.

```
int testScore;
   cout << "Enter your test score: ";
   cin >> testScore;
   if (testScore >= 90 )
      cout << "Your grade is an A" << endl;
   else if (testScore >= 80 )
      cout << "Your grade is a B" << endl;
   else if (testScore >= 70 )
      cout << "Your grade is a C" << endl;
   else if (testScore >= 60 )
      cout << "Your grade is a D" << endl;
   else
      cout << "Your grade is an F" << endl;
   return 0;
}
```

5 Compile, run and test your program.

PAUSE POINT Develop the program that you have just made in the Step by step by testing it against unwarranted data. For example, establish what would happen if you typed in erroneous data. The result would be incorrect. Develop the program further to validate against this happening.

Hint Loops can be used in validation. Research how loops can be used to stop erroneous data from being input to give incorrect results.

Extend Now see if you can develop the same program but, this time, use case statements instead of IF statements.

Function calls

You have already seen how programming languages have a set of predefined (built-in) functions (see the section on 'Built-in functions'). If a programmer makes their own functions rather than using built-in functions, they are referred to as 'custom-made' or 'user-defined' functions. In this section, we will explore custom-made functions further.

Link

For more about built-in functions, see the section on 'Built-in functions', in 'Programming paradigms'.

One of the main reasons why you may want to make your own functions is because, when creating a computer program, there are often sections of code that you may want to reuse or repeat. Chunks of code (a set of instructions) can be given a name and they are known as functions. Creating functions saves time because the program only has to execute (call) the function when it is required, instead of the programmer having to type the whole set of instructions again.

Defining functions

The structure for defining a function is shown in Figure 4.50.

```
return_type function_name( parameter list )
{
   body of the function
}
```

Figure 4.50: Structure for defining a function

Return type

A function may wish to return a value. The return type is the data type of the value that the function returns. Some functions perform the desired operations without returning a value. In this case, the return type is the keyword void.

Function name

The function name is the actual name of the function. It must be given a meaningful name so that it makes sense in relation to what the function will do. This will allow the programmer to remember what this function is for or for other programmers to use it.

Parameters

A parameter is like a placeholder. When a function is called, you pass a value to the parameter. This value is referred to as the 'actual parameter' or 'argument'. The parameter list refers to the type, order and number of the parameters of a function. It is possible that a function may contain no parameters.

Function body

The function body contains a collection of statements that define what the function will do.

Declaring functions

When a function is declared, it is telling the compiler about the function name and how to call the function. The structure of a function declaration is shown in Figure 4.51.

```
return_type function_name( parameter list );
```

Figure 4.51: Structure of a function declaration

You can define a function in one source file and call that function into use in another file.

Calling functions

When a program calls a function, the program control is transferred to the named function. A called function performs the defined task (which has been developed by the programmer), and when its return statement is executed or when its function-ending closing brace is reached, the function returns program control back to the main program. (Braces are { and }. The second of these is a closing brace.)

To call a function, you simply need to pass the required parameters, along with function name. If the function returns a value, then you can store that value.

Step by step: Declaring and calling a function in C++

7 Steps

1 Open C++ on your computer and on the first line type the standard library.

```
#include <iostream>
```

2 On the next line type, the following. (Remember: this will help you not to duplicate std when you want to use cout or cin.)

```
using namespace std;
```

3 To declare your function, type

```
// function declaration
    int max(int num1, int num2);
```

4 Start the program by typing

```
    int main()
    {
```

5 Declare your local variables by typing

```
// local variable declaration:
   int a = 500;
   int b = 1000;
   int result;
```

6 Call the function.

```
    {
// local variable declaration:
    int a = 500;
    int b = 1000;
    int result;

// calling a function to get max value.
    result = max(a, b);

    cout << "Max value is : " << result << endl;
    }
```

7 Provide the code for function max(), which returns the larger of the two numbers.

```
    // function returning the max between two numbers
int max(int num1, int num2)
{
    // local variable declaration
    int result;

    if (num1 > num2)
        result = num1;
    else
        result = num2;

    return result;
}
```

Data structures

Data structures are used in computer programming to store and process data. There are different types of data structure that a programmer can utilise, and some of these will be explored in this section.

Lists

A list is a basic data structure in which each item contains the information that the programmer needs in order to get to the next item on the list. The advantage of lists over arrays is that they provide a programmer with the capability to rearrange the list efficiently.

Figure 4.52 gives an example of using different functions to manipulate elements within a list.

```
#include <iostream>
#include <list>

int main()
{
    std::list<int> CreatedList;     //Create List called CreatedList

    CreatedList.push_front(1);      //Demonstration of member functions you can use using lists
    CreatedList.push_front(2);
    CreatedList.push_back(3);
    CreatedList.push_back(4);
    CreatedList.insert(CreatedList.begin(), 5); //Inserts new element at the beginning

    //therefore the list shoudl appear 5, 2, 1, 3, 4

    //for loop iterates through the list, outputting them in order from start to end
    for (std::list<int>::iterator num = CreatedList.begin(); num != CreatedList.end(); num++)
    {
        std::cout << *num << std::endl;
    }
    system("PAUSE");
    return 0;
}
```

Figure 4.52: A list

Arrays

In programming, an array is a structure that holds data of the same type. Single-dimensional arrays and multi-dimensional arrays are two different types of array that can be utilised in programming.

Single-dimensional arrays

A one-dimensional array (or single-dimensional array) is the simplest type of array, and contains only one row for storing data. It has a single set of square brackets [ ].

In C++, an array can be created by a simple line of code, as shown in Figure 4.53.

```
#include <iostream>

using namespace std;

int main()
{
int testscores[5] = { 10, 7, 5, 8, 4};      //Array is created here

cout << "1st member = " << testscores[0] << endl;   //Output the first element in the array
cout << "4th member = " << testscores[3] << endl;   //Output the third element in the array

return 0;
}
```

Figure 4.53: Array created in one simple line of code

In the example in Figure 4.53, an array called 'testscores' that will store five integers has been created. Notice how the values are added to the end of the declaration in the code. This means that the array will look as in Figure 4.54.

10	7	5	8	4

Figure 4.54: Array 'testscores' example

When creating an array, you must also consider that elements need to be numbered, with the first element (the left-most) being element '0', as shown in Figure 4.55.

0	1	2	3	4
10	7	5	8	4

Figure 4.55: Array 'testscores' showing left-most element being '0'.

Multi-dimensional arrays

The simplest form of a multi-dimensional array is a two-dimensional array. A two-dimensional array is, in effect, a list of one-dimensional arrays. The structure of a two-dimensional array is as shown in Figure 4.56.

```
type arrayName [ size1 ][ size2 ];
```

Figure 4.56: Structure of a two-dimensional array

A three-dimensional array is simply extended from a two-dimensional array, so that its structure is as shown in Figure 4.57.

```
type arrayName [ size1 ][ size2 ] [ size3 ];
```

Figure 4.57: Structure of a three-dimensional array

Records

A record is a form of data structure that allows a programmer to combine several data items of the same kind. In programming, arrays allow a programmer to define variables that combine several data items of the same kind. However, in C-derived programming languages, we can use 'struct' (record), which is a user-defined data type that allows programmers to combine data items of different kinds.

Link

For more about records, see 'Records' in the section on 'Handling data' in a program, in 'Programming paradigms'.

Sets

In programming, sets can be used as forms of data structure. Sets allow programmers to store elements only once. This allows items added to a set to appear on only one occasion. For example, if you have a list of multiple words, the word 'apple' could appear several times. However, if this was implemented within a set, then the word 'apple' would appear only once.

Sets are useful as they reduce the amount of checking for duplication that is needed. For example, suppose you want data regarding the reference numbers for the customers of a gym. The gym reference number gives you access to additional information about the gym member such as their personal details. A gym reference number is unique so should appear only once. However, within the gym's data, the gym reference number appears more than once because a user may have subscriptions to different facilities such as the gym, swimming pool, classes and so on. Therefore, to see the information, we can set up a loop to add all the information to a set. As a set only allows for items to be represented once, the repeated data is omitted, leaving just the one instance of personal information required.

Assessment practice 4.4

AO5

Using the pseudocode you developed in the Pause Point at the end of the section on 'Structured English' (pseudocode), in 'Standard methods and techniques to develop designed solutions', develop the code that would make the Loan calculator in your chosen programming language.

Tip: Remember that, when doing mathematical calculations, you must abide by BIDMAS (Brackets, Indices, Division, Multiplication, Addition, Subtraction), otherwise you will derive the incorrect answer.

Plan

- What is the task? What am I being asked to do?
- Am I aware of all the coding conventions I need to apply to my computer program?
- Which programming language should I use to develop my solution?

Do

- I know what it is that I am doing and what I want to achieve.
- I must make sure that I use correct naming conventions when programming my solution.
- Have I used functions to structure my code more efficiently?

Review

- I can explain each stage in the program.
- I can state that I have used the correct algorithms to derive the correct answer.
- Can I compile my program and run it with no errors?

E Evaluating a software development project

Throughout the main stages of the software development life cycle, it is important to ascertain whether each stage has been implemented appropriately and correctly. An important reason for doing this is to ensure that what you are delivering to the client meets their requirements and is of a high standard. It is also important to consider that there are likely to be things that did not go according to plan during the development pocess. You can spot these through evaluation. As a software developer, you must be adaptable and, crucially, when things do not go according to plan, you must be aware of your mistakes so that you can avoid them in the future.

Evaluation of design

Design documentation is developed during the software development process once all of the requirements have been elicited from the client/users. Once the design documentation has been completed, action must be taken to evaluate the designs. This helps to make sure that as few difficulties as possible are faced by the software developers when they are programming the solution. This stage is quite crucial because, without evaluation, any faults or omissions in the designs will not be identified before the start of the development. Most importantly, the design for the program may not correctly solve the problem that the software was meant to solve, or may not solve it in the best way.

Appropriate application of BCS symbols to the flow chart

When evaluating a software design solution, you must ask yourself the following two questions regarding the flow chart design.

Link

Remember that you must use standard British Computer Society (BCS) symbols when creating a flow chart. For more on the application of BCS symbols, see the section on 'Flow charts and use of standard symbol conventions'.

Is it easily understood and fully decomposed?

When you design a software solution, it is essential for it to be fully **decomposed** and that you understand the problem to be solved. If the solution has not been fully decomposed or understood, then this will lead to problems when a software developer tries to create the programming solution using this design.

Key term

Decomposed – when a complex problem is broken down into more manageable component parts.

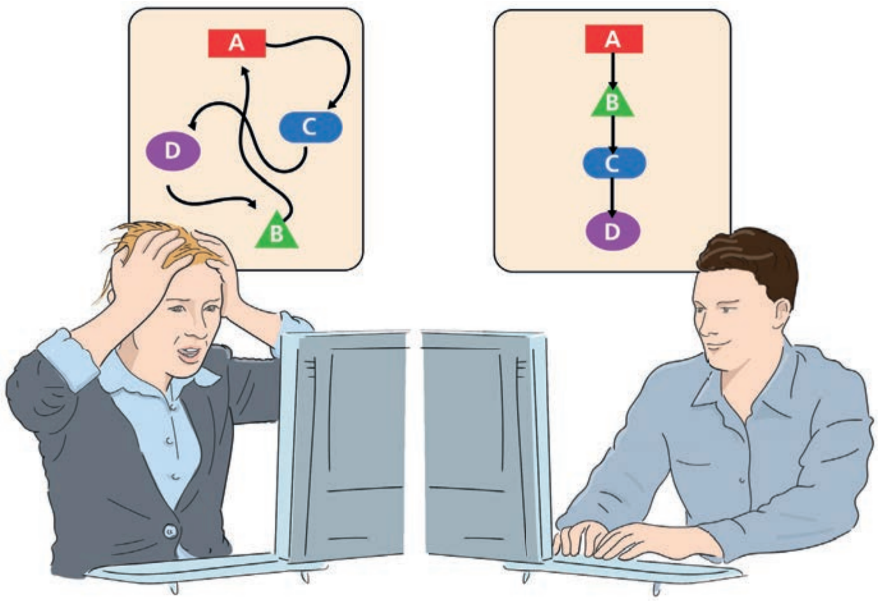


Figure 4.58: The solution must be fully decomposed

Consider Figure 4.59, which shows a flow chart for a snakes and ladders scenario. As you can see, the flow chart has not been fully decomposed, which means that when it is presented to a programmer they will have problems trying to develop the solution. For example, the flow chart has not been decomposed sufficiently to show the decision if you land on a snake or a ladder. It should be decomposed into two decisions which ask the questions: 'Have you landed on a snake?' and 'Have you landed on a ladder?' As it stands, another programmer would find it difficult to follow the functionality of what is required.

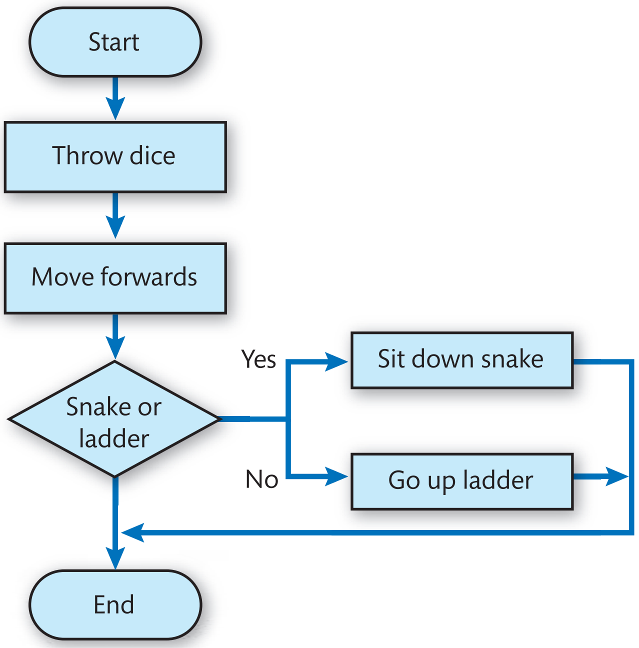


Figure 4.59: Poorly designed flow chart for snakes and ladders

Figure 4.60 shows how the snakes and ladders scenario should be presented in a flow chart that fully decomposes the problem that needs to be solved.

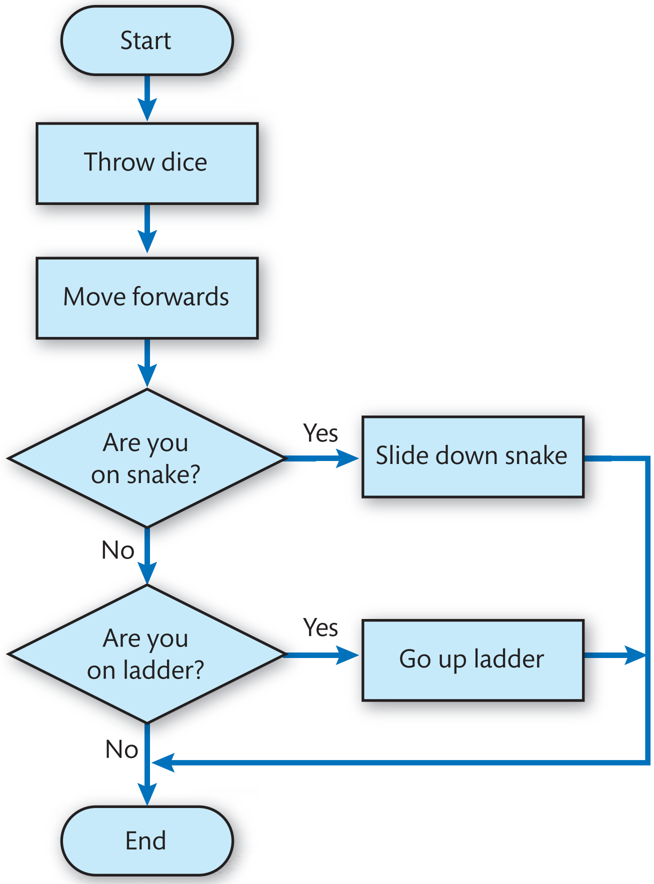


Figure 4.60: Well-designed flow chart for snakes and ladders

Essentially, the issue with the flow chart in Figure 4.59 is that it is missing an extra decision which acts as an IF statement (as shown in Figure 4.60). As the problem shown in Figure 4.59 has not been fully decomposed, an incorrect solution will be derived. However, in Figure 4.60, the problem **is** fully decomposed so the correct solution will be derived.

Therefore when evaluating your design, you must go back and ask yourself whether the flow chart decomposes the whole problem fully.

Is the flow chart complete?

You must also ascertain whether the flow chart is complete, that is, does it solve every aspect of the problem that was set out? In order to answer this, you should go through all the requirements that were specified by the client/users and cross-reference these with the flow chart to see whether they have all been covered.

Appropriate application of pseudocode

To ensure the appropriate application of pseudocode, you should evaluate it by asking the same questions as for the flow chart.

1 Is it easily understood and fully decomposed?
2 Is it complete?

These two critical questions about the pseudocode must be answered prior to the development of the program.

It is also important, when applying pseudocode, that it:

- should not be written in a specific coding language (but rather be in structured English)
- outlines the draft structure of the code you require
- is understandable.

Figure 4.61 shows an example of pseudocode which has not been decomposed properly. The weekly pay calculation has not been determined, which means that the programmer using this pseudocode to create the software solution would not know what the calculation will be. They could make an informed guess, but, if they were wrong, the program would give the wrong answer. Therefore, they need to go back to the designer and go over the omissions in the pseudocode to ensure that the problem is fully decomposed and solves every aspect of the problem. This would mean spending additional time and money during the implementation of the software development life cycle which would not have been accounted for in the project schedule or budget.

```
BEGIN
hourlypay = INPUT “How much do you earn per hour?”
hoursperweek = INPUT “How many hours per week do you work?”
IF weeklypay greater than 600 then
OUTPUT “afford to live alone”
ELSE
OUTPUT “cannot afford to live alone”
END
```

Figure 4.61: Poorly written pseudocode for a pay calculation

Figure 4.62 demonstrates how the pseudocode in the example shown in Figure 4.61 should have been written with the problem having been fully decomposed.

```
BEGIN
    hourlypay = INPUT "How much do you earn per hour?"
    hoursperweek = INPUT "How many hours a week do you work?"
    weeklypay = CALCULATE hourlypay multiple hoursperweek
    IF weeklypay greater than 600 then
      OUTPUT "afford to live alone"
    ELSE
      OUTPUT "cannot afford to live alone"
    END IF
END
```

Figure 4.62: Well-written pseudocode for the Figure 4.61 pay calculation

Coverage of functionality requirements and identifications of any omissions

When designing your software solutions, using either a flow chart or pseudocode, it is important to ensure that they cover all of the requirements set out during the analysis stage of the software development life cycle. The worked example here explains why it is important to ensure that all requirements are set out during analysis.

Therefore, when you evaluate your software designs, it is essential to ensure that the full range and coverage of all the requirements are stated in the design solution, as well as how these requirements will be fulfilled.

Worked example: Functionality requirements for baking a cake

If you were to bake a cake, on decomposing the problem, it would be necessary to know the following information (these are the requirements).

- What type of cake will you bake (e.g. Victoria sponge, chocolate or a fruit cake)?
- What equipment is required to bake the cake? (e.g. mixing bowl, utensils, scales and cake tin)
- What ingredients are needed and what quantity of each ingredient is needed? (i.e. the ingredients list)
- When and how to add each ingredient? (i.e. the method)
- For how long should the cake be baked?
- How many people will be eating the cake (i.e. how large a cake is needed)? This will have an impact on several other parts of the problem, for example the quantity of each ingredient that is needed, the size of the cake tin and for how long it will need to be baked.

In this example, part of the problem has been left out of the above information. The following have not been considered.

- Where will you bake the cake?
- At what temperature will you bake the cake?

If the initial list of information was used to create the cake, the recipe would say for how long the cake should be baked but it would not state that the cake should be placed in the oven or give the temperature of the oven. Even if the cake made it to the oven (because everyone would know to put a cake in the oven to bake), the likely outcome is that it would be under or over-baked because the correct baking temperature was not given in the recipe.

The issue with this solution occurred because the fact that the cake should be placed in the oven and the correct oven temperature were not specified (the problem had not been fully decomposed in the solution), making the solution (recipe) incomplete.

Identification of design strengths and potential further improvements to meet the scenario requirements

With any type of evaluation, it is important not only to identify design strengths but also to identify and explain any potential areas for further improvement. Details of these should be written in the design evaluation document. This provides the opportunity to improve the design at this point in the process or, if this is not possible due to constraints on the project, it allows the improvements to be noted as possible developments to be made once a program has been implemented, when further maintenance or upgrades are carried out.

For example, consider a program that requires users to log in. The login screen may work perfectly well, but there might be scope to add a help screen, which was not included in the initial design. The reason for this could be to save time and money in the initial development of the program. However, a help screen is something that the client may want to consider adding during future development because it would make the program more user-friendly and accessible.

Link

For more on the format and standard conventions of flow charts, see "Flow charts and use of standard symbol conventions', in 'Standard methods and techniques to develop designed solutions'.

Link

For more on pseudocode, see 'Structured English (pseudocode)', in 'Standard methods and techniques to develop designed solutions.'

The strengths and weaknesses of using a flow chart versus using pseudocode for a specific scenario and the reasons why

For any software design, both flow charts and pseudocode have their own strengths and weaknesses. Sometimes, these strengths and weaknesses can be quite subjective, in that they can purely depend on a person's preference for what they feel is the best design method to use.

As previously discussed, a flow chart is used to show an overview of a problem diagrammatically. It is a pictorial representation of how the end program will work, and it follows a standard format as outlined by the British Computer Society (BCS).

Similarly, pseudocode is a means of representing the steps that need to be taken to solve the problem without worrying about the syntax of a particular programming language. Unlike flow charts, pseudocode is created in a written format, rather than a pictorial one.

There are no absolute rules for writing pseudocode, in contrast to full programming languages, and so it can be written in ordinary English. However, some could argue that this is a disadvantage of pseudocode because flow charts follow a standard formatting convention which makes them easy to explain to other programmers.

The main advantage of pseudocode over flow charts is that it is actually very similar to the final program code. Pseudocode requires less time and space to develop than a flow chart and can be written in your own way because there are no fixed rules. However, flow charts are capable of showing the overall flow of instructions from one process to another, which is something that pseudocode does not do. Moreover, with a flow chart, you can see the individual processes at a glance (for example, the number of decision-making operations). It is also easier to show iteration and conditional statements using a flow chart, whereas, with pseudocode, these can easily be as complex as in the program code itself. Table 4.10 summarises some of the main advantages and disadvantages of flow charts and pseudocode as design methods for software.

When working in the computing industry, it is highly likely that you would be asked to design a programming solution. As you can see, each design method has its own advantages and disadvantages. Quite often, it is normal to use both of these design methods, and both would be presented to a team of programmers who would develop a software solution from them.

Table 4.10: Advantages and disadvantages of flow charts and pseudocode

Design method	Advantages	Disadvantages
Flow chart	• Standardised approach. • Visual appeal.	• Can be hard to modify depending on what software you use. • Special software is required e.g. Microsoft® Visio.
Pseudocode	• Easily modified. • Structured concepts (design elements) are implemented. • Can be created easily in a word processor or text editor.	• There is no accepted standard and so it varies from programmer to programmer. • Not visual and can be hard to understand as an overview of the program.

Evaluation of software testing

Testing is not a straightforward task. It requires careful planning to ensure that the software program is robust, gives the correct output and meets the requirements that it was designed to fulfil. A wide range of tests to ensure that every aspect of the performance of the software is assessed will have been devised during the design phase of the software development process. Once the software program has been created, you can proceed with the testing stage, which involves going back to the test plan, performing the tests and recording the actual outcomes.

Link

For more about normal, abnormal and extreme test data, see 'Choice of test data', in the section on 'Test data', in 'Standard methods and techniques to develop designed solutions'.

Different types of testing

When developing a test plan for your software, it is essential that the tests you subject it to cover a wide range of data. Your tests should cover normal, abnormal and extreme test data. Consider the consequences if you failed to utilise a wide range of tests. Would an unexpected bug be a minor inconvenience or would the result be more serious?

Link

For more about test plans, see 'Expected test results from a range of testing methods', in the section on 'Test data', in 'Standard methods and techniques to develop designed solutions'.

Case study

Testing the Patriot Missile Defence System

In 1991, the Patriot Missile Defence System was developed by the United States government. This system was proposed by President Ronald Reagan and was first used in the First Gulf War as a defence against Iraqi Scud missiles. News agencies reported that it was a resounding success. What they failed to mention was that several missiles broke through the defence shield, one of which killed 28 US soldiers. In-depth analysis of the system determined that a 'bug' or a software error was responsible. A timing error in the system's clocks accumulated so that, after 14 hours, the system was no longer accurate. What conclusions can be drawn from this? One reason given was that the defence system was not tested effectively enough.

Some software bugs can be merely inconvenient, but others can be costly, or even catastrophic. The case study above encompassed all of these traits, as it not only proved costly and time consuming to fix, but resulted in the loss of life.

Check your knowledge

1. What was the overall result of the failure, aside from the loss of life?
2. What form of testing should take place on similar projects to ensure that this does not happen again?
3. Who was ultimately responsible for failing to test the system effectively?
4. How long should you commit to testing a software program to ensure that is safe or effective?
5. What other examples of testing failures have resulted in the loss of life?

Tip

As mentioned previously, you should repeat all of the tests – not just the ones that failed the first time – because, by fixing problems highlighted by the failed tests, it is possible that you may have introduced an error elsewhere in the program.

Tip

Highlight failed tests in a different colour to passed tests. This will make it easy to identify them.

Reflect

Reflect on the questions asked at the beginning of this section in the light of the case study on the testing of the Patriot Missile Defence System. If you failed to utilise a range of tests to effectively test your software, what would the consequences be? Would an unexpected bug be merely a minor inconvenience? Would the consequences be more significant than that?

Recording of actual results and analysis

Once a software program has been developed, the test plan should be revisited and the tests performed. The actual results of the tests should be recorded. These actual results should then be compared with the expected results. It is extremely likely you will come across some tests that do not provide the expected results. Any such instances need to be documented, and the issues should be fixed. The tests should then be repeated until the problems have been resolved by the programmer. Table 4.11 shows exemplar test data for a basic login system program.

Table 4.11: Test plan with exemplar test data for a basic login system

Test number	Purpose of test	Test data	Expected results	Actual results	Comments
1	To check that, when the correct username and password are input, the program accepts this.	Username: testing Password: Tim123!	The program should inform the user that they have successfully logged in and give them access to the program.	FAILED: The program generated a run time error and did not work. Please see screenshot 1 of the error	See screenshot 1. To be actioned.
1B	To check that when the correct username and password are input, the program accepts this.	Username: testing Password: Tim123!	The program should inform the user that they have successfully logged in and give them access to the program.	PASSED: The program worked as expected.	See screenshot 1B. A semicolon was omitted from line 23 which prevented the login system from working.
2	To validate the program to ensure that false information cannot be entered.	Username: Qqqqqqqq Password: 1111111111	The program should inform the user that the incorrect username and password has been entered.	PASSED: The program worked as expected.	See screenshot 2
3	To validate the program to ensure that false information cannot be entered.	Username: Testing Password: Tim123!	The program should inform the user that the incorrect username and password has been entered.	PASSED: The program worked as expected.	See screenshot 3
4	To validate the program to ensure that false information cannot be entered.	Username: Testing Password: tim123	The program should inform the user that the incorrect username and password has been entered.	PASSED: The program worked as expected.	See screenshot 4

Test 1 has been repeated

Discussion

In small groups, discuss which of the tests in Table 4.11 are normal, abnormal and extreme.

Commenting on results

It is important that test failures are clearly identified and that the nature of each error is highlighted. This can be done by taking a screenshot to show precisely what error message appeared. This information can be taken back to the programmer who can then perform remedial action to fix the error. When remedial action has been taken, it is important that the actions taken to overcome the error are documented in the comments column. These comments provide an audit trail of the errors as well as the steps that were taken to overcome them. Test number 1B in Table 4.11 provides an example of this.

Test records

It is crucial that you keep records of what tests have been performed and that you document how you have completed the tests and the methods you have used.

Completion of test records

It is important that the full range of tests that were devised should be completed to ensure the accuracy and robustness of the software program. Therefore, each test needs to be commented upon appropriately and, when a test has passed, this should be commented upon in the actual results. Sometimes, it is beneficial to emphasise that the test has passed by labelling it in the actual results column with some commentary.

Conversely, when there has been a test failure, this also needs to be identified. Moreover, remedial action needs to be taken to ensure that the identified error is resolved. The tests should be repeated until the desired outcome (expected result) is achieved.

Taking of and storing screenshots of tests

Whenever you perform a test, whether it passes or fails, it is important to document the process and prove the result in the form of a screenshot. This evidence demonstrates to anyone who is part of the software development process that each test has been acted upon appropriately.

Whenever you take a screenshot of your program, it should be itemised so that you know which screenshot corresponds to which test. Therefore it is important to store your screenshots in a folder, with each file name corresponding to the test in question.

Tip

Microsoft Windows® 7 and 10 operating systems both have a snipping tool which can highlight precisely what you want to copy from your screen. These can then be saved as picture files.

Research

Research what 'white hat hackers' are and see how they are used by organisations.

Making use of testing outcomes

It is important that all tests have a desirable outcome: that is, they should achieve the expected result. If there is a test failure, then it is the responsibility of the programmer to put this right. Once all testing outcomes have been resolved, it is then possible to analyse the data to discover what errors occurred and how these were ultimately resolved. This process is important as it can help a programmer to develop strategies to ensure that these desirable testing outcomes are reached more easily on subsequent software development projects.

Using iterative processes to improve accuracy, readability and robustness

An iterative process is one in which something is continually repeated in a systematic way until the desired outcome is reached. In software development, the iterative process views the software life cycle as a continuous loop.

The iterative process of software development begins by implementing a small set of requirements, which continually evolve in a loop in the software development life cycle until the complete software system is implemented and ready to be deployed. Therefore, the process does not start with a full specification of requirements. Simply put, the iterative process begins by specifying and implementing part of the requirements for the software rather than all of the requirements at once. The resulting software is tested and then further requirements are implemented and tested until the software is fully complete and fit for purpose.

The advantage of adopting this iterative process is that a working piece of software, with some of the key requirements, is ready much earlier in the development process. It would be possible to release this early version of the software to customers in order to get feedback on which elements customers like and do not like. It also makes it easier to discover design flaws and, because errors and issues are identified early in the testing process, this means that the end program will be more accurate, readable and robust.

This could take several cycles of testing and redevelopment of code until the desired outcome is reached. This iterative process ultimately helps to develop a program that is robust, as the error will have been resolved through code development.

However, a program may derive a correct output, but this does not necessarily mean that the program is accurate. Assume that you have a program that utilises an algorithm that determines the speed of a moving car. Just because one correct answer in your testing has been derived, it does not necessarily mean that the program is accurate. Therefore an iterative process is used to ensure that the program derives the correct answer multiple times. Different sets of data will need to be used to test the program several times to ensure that the right result is given every time. Once the correct output has been successfully achieved a number of times, you can be certain that the program is accurate.

The same process also applies to readability. You need to be certain that the output derived from the program makes sense, is accurate and is readable. Multiple tests should be carried out in an iterative process to ensure that this is the case.

Link

For more about readability and robustness, see 'Factors that contribute to the quality of code', in the section on 'Code readability', in 'Software design considerations'.

Determining which tests were successfully met and which test data issues were not resolved

When completing a test plan using black box (functionality) testing, it may be that not all the tests are successfully passed first time. Therefore it is important to record what the errors were and the methods that were used to fix the problems. You can achieve this by recording the issues and how you resolved them within the comments section of your test plan. By recording this information clearly in your test plan, other programmers in the software development team will be able to see what has already been tried in order to fix a particular problem, which means that they will save time by not trying the same fixes to the problems again.

Link

For more about black box testing, see 'Functionality testing', in 'What to test', in the section on 'Test data', in 'Standard methods and techniques to develop designed solutions'.

Identifying own learning and skill requirements arising from the testing process

As a programmer, it is important to reflect on and evaluate your own learning and skill requirements. When you test a program, it may be that you have successfully completed all the tests and solved all the errors. However, within the computing industry, this is a luxury that will not happen all the time. Quite often you will be challenged with problems to which you may not necessarily have an immediate answer. However, these problems can be overcome with patience, perseverance and by working in a team with other software developers. You should not expect to always have the answer yourself, in isolation from your team. Therefore, it is important to reflect upon what you have learnt from the testing process and what skills you have developed. By evaluating this process for yourself, you will be able to learn and develop as a programmer much faster.

PAUSE POINT

Download the sample assessment material for *Unit 4: Software Design and Development Project* from the Pearson PLC Qualifications website: https://qualifications.pearson.com/.

In this sample assessment material, you are challenged to develop a BMR calculator for a gym. Using this scenario, develop a series of tests for the software, using a range of test data. Working on your own or in small teams, develop a solution to this scenario in your preferred programming language, and use the test data to record the actual outcomes.

Hint Remember to save your test outcome screenshots in a separate folder and to name the files so that they correspond with the correct test.

Extend Once you have completed the program in the language of your choice, consider how it could be implemented in another language. For example, if you initially used C++, then explore what the solution would look like in Python®.

Evaluation of the software

When a piece of software has been developed, it is important to ensure that it meets the requirements of the brief and the expectations of the client and potential users.

Strengths and weaknesses of the software

Whenever you create a software solution, it is important to identify if you were successful in what you set out to do. Part of this process involves identifying the strengths and weaknesses of the software. The following section looks at the various aspects you need to address when appraising the strengths and weaknesses of software.

Solution fitness for purpose

Once a software solution has been developed, it will need to be presented to the client, so that they, as well as the developers, can analyse whether the solution that has been implemented is **fit for purpose**. This is an opportunity for the client to highlight strengths of the software as well as weaknesses. If there are any serious issues with the software that the client is not happy about, then this is an opportunity to go back and make any necessary refinements to it. Whether the software is fit for purpose in any particular regard is something that needs to be agreed between the client and the developers. Any refinements that are necessary to make the software fit for purpose will need to be completed before the software can be installed, otherwise there will be problems when it is used.

Key term

Fit for purpose – a term used to ascertain whether the end product (in this case the software) meets the requirements of the client.

Intuitiveness and ease of use

It is essential that the resulting software program is intuitive. An intuitive software program is one that a user can understand how to use without having to think too much. For example, let us assume that a programmer has developed a login program, but the username and password boxes were not the right way around. We assume the username types in the username is first, and password last. If it was not the right way around, it would be harder to understand how the program works, and it would not be intuitive.

The best way to ensure that a software program is easy to use and intuitive is to get potential users to evaluate the software and then to make any necessary refinements before it is released. It would be sensible, in order to save time and money, to get potential users to evaluate the design of the software at an early iteration stage of the software development process, so that their feedback can be acted upon before the software is completed.

Constraints of the programming language used

All programming languages are different, and every programmer has a preference for which language they would like to use. Some programmers prefer C-derived family programming languages, such as C++, while others prefer languages such as Python®. To say which one is best would be very subjective, that is, a matter of opinion. Instead, it is best to consider the strengths and constraints (or limitations) of each programming language. For example, you might have noticed that Python® needs substantially less code than C++.for the same program, which, in a sense, makes it easier to use. C++ enforces strict restrictions when defining data such as integers and characters. This is useful, as the compiler prevents you from mixing different kinds of data together, which helps avoid programming errors.

The decision about which programming language you will use should be informed by your knowledge of the strengths and constraints of each language, the purpose of your program and any outside constraints put on you, for example the requirements of your client.

Maintainability of the program

Once a software program has been implemented and evaluated, it needs to be installed. When it is first installed, it will be monitored to check whether or not it is working correctly. Sometimes, problems with a software program will not be found until it is being used by a large number of people or until an unusual situation occurs.

The software program should be maintained by analysing the performance of the program in use by means of various evaluation techniques, such as interviewing clients and users or giving them questionnaires.

Link

For more information about maintainability of code, see 'Maintainability', in the section on 'Code readability', in 'Software design considerations'.

Extent to which the software meets the requirements of the brief

The crucial part of evaluating software is to establish the extent to which the software fulfils all of the requirements of the brief. You can do this by creating a table in which all the requirements are stated and then carrying out a full appraisal as to whether each requirement has been met. The client, as well as the team of software designers and developers, should be involved in this appraisal.

Maintenance of the software

Even when a software system is up and running correctly, it can still require maintenance. This may be because a bug is found, because the organisation needs a new feature to be added to the software, or it might be because of changes to external but connected software. For example, consider the purchase of a car. A car is tested annually in the form of an MOT (Ministry of Transport) test to ensure that it still works as it should. You may also wish to upgrade the car or install a security alarm. The same is true of software; after a period of time further developments or updates may be required to the software.

Link

For more about maintenance, see 'High maintainability', in 'Selecting and applying common good practice design concepts', in the section on 'Design concepts, in Software design considerations'.

Identifying own learning and skill requirements arising from the software development process

Throughout the whole software development process from the conception to evaluation, you will have developed your own learning and skills. It is likely that you will have had moments at the start of the development of the solution to a complex problem using a programming language when you did not know how to proceed. In this situation what did you do? How did you overcome the problem? What did you learn about yourself? For example, did you learn to how to be patient or how to persevere in the face of adversity?

Ultimately, programmers are problem solvers, and problem solvers research the problem and think creatively to devise a solution to the problem. As you undertake a project, you are continually developing and learning. The skills you learn will ultimately help you to become a better and more skilled programmer.

Assessment practice 4.5

A04

Using the software program that you developed for the loan calculator in Assessment practice 4.4, as well as the pseudocode that helped design it, you are to write an evaluation that covers the following.

1. The quality and performance of your program.
2. The choices you made about coding conventions.
3. The challenges you faced and how you overcame them.
4. What would you do differently if you were to do the task again?
5. How did the pseudocode influence your program?

Give yourself 2 hours to complete this task, to replicate the controlled conditions of the live assessment.

Plan

- What is the task? What am I being asked to do?
- I will refer back to previously saved work, and make a note of what I did.
- I will make a note of each of the five points listed, highlighting any areas of significance.

Do

- I know what it is that I am doing and what I want to achieve.
- I can write effectively about each point and can identify not only areas of strength, but developmental areas too.

Review

- I can self-evaluate the progress that I have made.
- I can work under strictly timed conditions to complete a task.

Further reading and resources

Textbooks

Davis S – *C++ for Dummies, 7th Edition* (John Wiley & Sons, 2014)

Green J – *Head first C#, 3rd Edition* (O'Reilly Media, 2013)

McGrath M – *Python in easy steps, 1st Edition* (In Easy Steps Ltd, 2013)

Websites

- www.cplusplus.com/doc/tutorial/
- www.python.org/
- https://www.youtube.com/watch?v=tvC1WCdV1XU&list=PLAE85DE8440AA6B83 – This is the first of 73 video tutorials on how to use C++.
- https://www.youtube.com/watch?v=tvC1WCdV1XU&list=PLAE85DE8440AA6B83 – This is very useful to anyone wishing to learn C++ programming who is inexperienced at using it.
- https://www.youtube.com/watch?v=4Mf0h3HphEA&list=PLEA1FEF17E1E5C0DA – This is the first of 43 video tutorials on how to use Python® programming. These are produced by the author 'Bucky', who is adept at explaining how to programme in Python®.
- http://www.codeacademy.com – This is useful to anyone wishing to learn Python® programming who has little or no experience of how to use it.

THINK FUTURE

Gary Chapman

Senior Software Developer

I've been working in software development for over seven years, since finishing my BTEC National Diploma in IT. I was fortunate enough to get an apprenticeship with a local software development company, and now I have worked my way up in the company to be a senior software developer. What amazes me is that every single day I am learning something new. When I first started, I was developing programs in C, but now this has developed to using software languages such as C# and Python®. I have become accustomed to thinking on my feet and developing innovative solutions to complex problems.

People are often surprised that I work in a large team of software developers. There is a perception that in software development it is your sole responsibility to develop a solution. This is far from the case. The majority of my time involves working and communicating with software designers to develop the most efficient and effective solutions possible. We do this so that our customers are happy with the end results. I also have to liaise and work with software testers to ensure that what I have developed with my team is fit for purpose.

As a software developer, I have learnt that it is not just about developing the technical programming skills. It is also about learning how to communicate with a wide range of audiences. I have realised that if I don't communicate effectively with my customers and the end users, then the end result will not be fit for purpose. You need a whole range of skills to be a software developer. The best thing is that each day is different, and this work gives an opportunity to work in a positive, team environment in which we develop effective solutions to problems.

Focusing your skills

Designing a software solution

It is important to know how to apply correct and accurate designs. This is essential because, without deriving the correct requirements for a design, this would lead to an incorrect solution being developed. Therefore you need consider these questions.

- Have you managed to elicit all the requirements from the client and documented them fully?
- Have you applied the correct symbol conventions to your flow chart, and does it promote a correct solution?
- Is your test plan robust to ensure that your program is subjected to a range of test data?
- Is your pseudocode readable, allowing a programmer to develop a programming solution?

Developing a software solution

Once your designs and test plan has been completed, you will need to develop a software solution within either a C-derived programming language or Python®. When developing your software solution, you should consider these questions.

- When developing your programmable solution, have you got your flow chart and pseudocode with you?
- Have you commented on your code throughout?
- Have you indented your code to make it more readable?
- Have you used correct naming conventions throughout, for example when you create functions or variable names?

Getting ready for assessment

This section has been written to help you to do your best when you take the external test. Read through it carefully and ask your tutor if there is anything that you are still not sure about.

About the test

This unit is assessed under supervised conditions and the number of marks for the unit is 68. Pearson PLC sets and marks the task.

The external assessment will last for 6 hours in a 1 week period which can be arranged over a number of sessions. You will be assessed on your ability to design, develop, test and evaluate a software program that meets the requirements of the client in the provided scenario.

The test is split into five main sections that all need to be completed.

- Activity 1: Flow chart.
- Activity 2: Pseudocode.
- Activity 3: Test Log.
- Activity 4: Program.
- Activity 5: Evaluation.

This assessment does not involve multiple-choice questions but, instead, will be based on your ability to analyse a scenario to design, test, implement and evaluate a programmable software solution.

Each activity will state how many marks are on offer for its completion. You should attempt to complete all of the activities.

Sitting the test

The test should be carried out under supervised conditions.

- Electronic templates for use in activities 3, 4 and 5 will be provided for centres to download for you to use ahead of your assessment. These will be supplied to you at the start of your assessment.
- Work should be completed on a computer. Ensure that you have a laptop lead if you are using a laptop.
- Internet access is not permitted.
- The task must be completed using one of following programming languages: a C family language or Python® 3.4 (or later version).
- Access to a data dictionary for the chosen language is permitted.
- You may have access to a calculator on your computer.
- During any break, materials must be kept securely by your tutor.
- You must not bring anything into the supervised environment or take anything out without your tutor's knowledge and approval.
- You should make sure that you back up your work regularly. You should save your work to your folder using the naming instructions that will be indicated in each activity.

Do not forget anything else you might need, for example glasses for working onscreen and refreshments.

Make sure that you arrive in good time for each test session and check that you have everything you need for the test ahead of time.

Plan out your work to ensure that you leave yourself enough time to check through your answers. Proofread and correct any mistakes before handing in your work. Ensure that you have looked at all sides of the assessment task sheets before starting.

Listen to, and read carefully, any instructions you are given. Lots of marks are often lost through not reading instructions properly and misunderstanding what you are being asked to do.

Key terms typically used in assessment

There are some key terms that may appear in your assessment. Understanding what these words mean will help you to understand what you are being asked to do.

- The following table shows you the key terms that will be used consistently in your assessments to ensure that you are rewarded for demonstrating the necessary skills.
- Please note: the list below will not necessarily be used in every paper/session and is provided for guidance only. Only a single command word will be used per item in your test.

Key terms	Definition
Annotation	A term used in computer programming to refer to documentation and comments that may be found on code logic. It can provide the rationale behind the logic or an explanation of how the logic accomplishes its purpose or goal for those who are to use the code or modify the code at a later date.
C family	A family of programming languages that includes all languages that are descendants of the C programming language.
Coding conventions	Guidelines for a programming language that recommend programming style, practices and methods for each aspect of a piece of program written in this language. It may cover: file organisation, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc.
Evaluate	A review and synthesis of each stage of software design and development processes and outcomes to provide a supported judgement about the quality. Typically, a conclusion will be required.
Flow chart	A formalised graphic representation to show the logic sequence of the program and define relationships.
Logical operators	Used primarily to determine the flow of a program through the use of selection (if statements and iteration (looping)).
Program	A list of instructions that tell a computer what to do. It is also used to refer to the software product that is provided to meet a client's brief.
Pseudocode	An informal high-level description of the operating principle of a computer program or other algorithm. It uses the structural conventions of a programming language, but is intended for human reading rather than machine reading.
Python® 3.4 or later version	Python® is a programming language. For the purposes of learner assessment, the version of Python® used must be 3.4 or a later version.
Quality of a program	The reliability, robustness, usability, efficiency/performance and maintainability of a software product.
Test data	Data that has been specifically identified for use in the testing of a program.
Test log	Used to plan and record program testing by recording the outcomes of testing and the changes made to solve problems.

Sample answers

Look at the sample questions which follow and the tips on how to answer them well.

Worked example

Set task brief: ShapeApp

You have recently applied for a programming position at a local company. The response to the job advert role was great and, although you were successful in your written application, the company would now like to gauge your software development skills. This will enable the company to restrict the number of candidates they invite to interview, as they will know that they have selected the best programmers.

The company has presented you with a problem to solve. They require you to design, develop and test software which has been developed by you, which is capable of performing calculations based on particular attributes of shapes derived by a user's input.

The program will allow the user to select what shape and attribute they want to calculate. The program will then ask for the necessary inputs from the user to calculate the attribute.

For example, calculating the Area (attribute) of a Rectangle (shape) based on the Height and Width (user's input).

The following shapes and attributes need to be calculated by your software program. It is expected that the formulae for these shapes are researched by you, if you do not know them already.

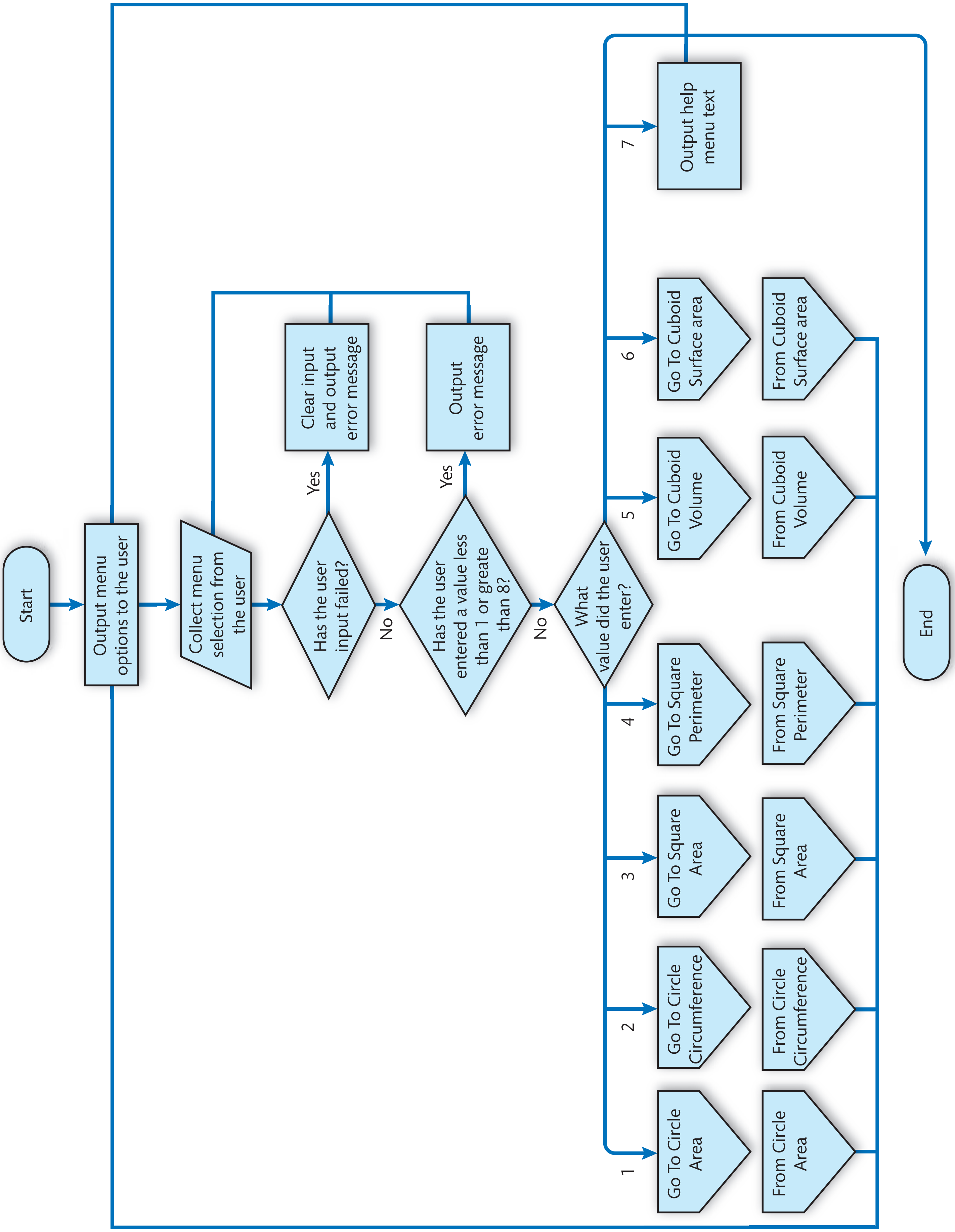
The program must contain features that validate all user input to ensure that extreme values and invalid inputs are not processed. The program must not shut down unless the user wants to quit; any closing of the program without a user initiating a close will be considered a software crash. There must also be a help menu which is informative and that explains how to use the program.

Activity 1. Produce a flow chart, using British Computing Society symbols, to plan the logic and processes for the program. [10 marks].

The question uses the command word 'produce', which means that you do not have to explain anything or write a report. You are required to produce (create) a flow chart using a suitable software package. Microsoft® Visio is ideally suited to this, but, if this package is not available, any Microsoft® Office package can do this by going to: Insert, **Shape, Flowchart**.

Sample answer: See the ShapeApp flowchart.

Shapes and attributes	Formula
Circle – Area	A = πr^2
Circle – Circumference	C = $2\pi r$
Square – Area	A = L^2W
Square – Perimeter	P = 4L
Cuboid – Volume	V = LWH
Cuboid – Surface Area	A = 2WL + 2LH + 2HW Or Area = 2 × Width × Length + 2 × Length × Height + 2 × Width × Height


Start
Output menu options to the user
Collect menu selection from the user
Has the user input failed?
Yes
Clear input and output error message
No
Has the user entered a value less than 1 or greate than 8?
Yes
Output error message
No
What value did the user enter?
1
Go To Circle Area
From Circle Area
2
Go To Circle Circumference
From Circle Circumference
3
Go To Square Area
From Square Area
4
Go To Square Perimeter
From Square Perimeter
5
Go To Cuboid Volume
From Cuboid Volume
6
Go To Cuboid Surface area
From Cuboid Surface area
7
Output help menu text
End

Activity 2. Produce pseudocode that a software developer could use to create the program. [10 marks].

As with the previous question, you are not being asked to explain anything. Instead you are being asked to produce something. Develop the pseudocode using a basic word processing package, and remember the conventions you learnt about pseudocode.

Sample answer: ShapeApp pseudocode.

```
Include iostream library

Declare openMainMenu procedure
Declare openHelpMenu procedure
Declare exitProgram procedure
Declare calcCircleArea procedure
Declare calcCircleCircumference procedure
Declare calcSquareArea procedure
Declare calcSquarePerimeter procedure
Declare calcCuboidVolume procedure
Declare calcCuboidSurfaceArea procedure
Declare collectLength double function that will parameter pass by reference
length
Declare collectWidth double function that will parameter pass by reference width
Declare collectDepth double function that will parameter pass by reference depth

Implement main function as return type integer
   Call openMainMenu procedure
   Return 0
Implement openMainMenu procedure as return type void
   Declare integer data type variable shapeselection initialised to 0
Output message “Please choose from the following options: Press 1 to calculate
a Circle Area, Press 2 to calculate a Circle Circumference, Press 3 to calculate
a Square Area, Press 4 to calculate a Square Perimeter, Press 5 to calculate a
Cuboid Volume, Press 6 to calculate a Cuboid Surface Area, Press 7 to access the
help menu, Press 8 to exit the program”
Collect input from user and store in shapeselection
If user input fails
   Clear user input
   Ignore user input
   Output message “Input error. Integer values only”
   Call openMainMenu procedure
Else if shapeselection is less than 1 or greater than 8
   Output message “Input error. Selection not valid”
   Call openMainMenu procedure
Else
   Switch statement with variable shapeselection
   If case is 1
      Call calcCircleArea procedure
   If case is 2
      Call calcCircleCircumference procedure
   If case is 3
      Call calcSquareArea procedure
   If case is 4
      Call calcSquarePerimeter procedure
   If case is 5
      Call calcCuboidVolume procedure
   If case is 6
      Call calcCuboidSurfaceArea procedure
   If case is 7
      Call openHelpMenu procedure
   If case is 8
      Call exitProgram procedure
   If case is default
      Output message “invalid selection”
      Call openMainMenu procedure
```

Remember to indent your pseudocode and to structure your code to show clarity. Notice how there is a hierarchy of functions in the pseudocode that are declared and used consistently throughout. This is required in order to attain high grades.

Observe how the pseudocode uses appropriate naming conventions throughout, including helpful names for the variables and functions declared. This must be done consistently throughout your design, otherwise you will lose marks.

Notice how the operators used to perform the mathematical calculations are all correct. Your pseudocode needs to be precise in order to attain high grades. Failure to do this consistently will lose you marks.

```
Implement calcCircleArea procedure as return type void
   Declare double data type variable radius initialised to 0.0
   Call collectRadius and pass parameter radius
Declare double data type variable area and initialise it to 3.1416 multiplied by
radius multiplied by radius
   Output message “Circle area = “ and output variable area
   Pause the program
   Call openMainMenu procedure

Implement calcCircleCircumference procedure as return type void
   Declare double data type variable radius initialised to 0.0
   Call collectRadius and pass parameter radius
Declare double data type variable circumference and initialise it to 3.1416
multiplied by radius multiplied by 2
   Output message “Circle circumference = “ and output variable circumference
   Pause the program
   Call openMainMenu procedure

Implement collectRadius function as return type double with passed parameter
radius
   Output message “please enter the radius of the Circle”
   Collect input from user and store in radius
If user input fails
   Clear user input
   Ignore user input
   Output message “Input error. Double values only”
   Call collectRadius function and pass radius
Else if radius is less than 0
   Output message “Input error. Negative values are not valid”
   Call collectRadius function and pass radius
Else
   Return radius
Return radius

Implement calcSquareArea procedure as return type void

   Declare double data type variable length initialised to 0.0
   Call collectLength and pass parameter length
Declare double data type variable area and initialise it to length multiplied by
length
   Output message “Square Area = “ and output variable area
   Pause the program
   Call openMainMenu procedure

Implement calcSquarePerimeter procedure as return type void

   Declare double data type variable length initialised to 0.0
   Call collectLength and pass parameter length
Declare double data type variable perimeter and initialise it to length
multiplied by 4
   Output message “Square perimeter = “ and output variable length
   Pause the program
   Call openMainMenu procedure

Implement calcCuboidVolume procedure as return type void
   Declare double data type variable length initialised to 0.0
Declare double data type variable width initialised to 0.0
Declare double data type variable depth initialised to 0.0
Call collectLength and pass parameter length
Call collectWidth and pass parameter width
Call collectDepth and pass parameter depth
Declare double data type variable volume and initialise it to length multiplied
by width multiplied by depth
   Output message “Cuboid volume = “ and output variable volume
   Pause the program
   Call openMainMenu procedure
```

```
Implement calcSurfaceArea procedure as return type void
Declare double data type variable length initialised to 0.0
Declare double data type variable width initialised to 0.0
Declare double data type variable depth initialised to 0.0
Call collectLength and pass parameter length
Call collectWidth and pass parameter width
Call collectDepth and pass parameter depth
Declare double data type variable surfacearea and initialise it to length *
width * 2 + length * depth * 2 + width * depth * 2
   Output message “Cuboid surface area = “ and output variable volume
   Pause the program
   Call openMainMenu procedure

Implement collectLength function as return type double with passed parameter
length
   Output message “Please enter the length of the square or cuboid”
   Collect input from user and store in length
If user input fails
   Clear user input
   Ignore user input
   Output message “Input error. Numeric values only”
   Call collectLength function and pass length
Else if length is less than 0
   Output message “Input error. Negative values are not valid”
   Call collectLength function and pass length
Else
   Return length
Return length

Implement collectWidth function as return type double with passed parameter
width
   Output message “Please enter the width of the cuboid”
   Collect input from user and store in width
If user input fails
   Clear user input
   Ignore user input
   Output message “Input error. Numeric values only”
   Call collectWidth function and pass width
Else if width is less than 0
   Output message “Input error. Negative values are not valid”
   Call collectWidth function and pass width
Else
   Return width
Return width

Implement collectDepth function as return type double with passed parameter
depth
   Output message “Please enter the depth of the cuboid”
   Collect input from user and store in depth
If user input fails
   Clear user input
   Ignore user input
   Output message “Input error. Numeric values only”
   Call collectDepth function and pass depth
Else if depth is less than 0
   Output message “Input error. Negative values are not valid”
   Call collectDepth function and pass depth
Else
   Return depth
Return depth

Implement openHelpMenu procedure as return type void
   Output message “This is the help menu”
   Call openMainMenu function

Implement exitProgram procedure as return type void
Exit program
```

Activity 3. Produce a test log to plan the testing of your complete program, including test data and expected result. Show your test planning by completing the test log section of the given 'Test log and evaluation' document. [6 marks].

To attain top marks, you must demonstrate that your test plan is thorough and includes a range of normal, abnormal and extreme data. You also have to be accurate with your expected results and ensure that the correct result is derived. Notice how tests 4 and 5 are abnormal tests, 6, 8 and 9 are extreme tests and the rest are normal tests. This must be demonstrated throughout for all the shapes in the software ShapeApp.

You will be presented with a template of a test plan. Your task is to populate this test plan, based on data that will be input into the program. Remember to devise a range of tests to ensure that your program is fit for purpose and is robust.

Test Number	Purpose of Test	Test Data	Expected Results	Actual Results	Comments
1	Output menu information regarding calculator program.	Start program	As soon as program runs, a list of options should be presented, asking the user to type in a different number corresponding to the different options.		
2	To see if program will exit when selected.	Press 8	The program should exit when 8 is pressed.		
3	To see if program displays the help menu when selected.	Press 7	The program should present a basic help menu when 7 is pressed.		
4	To see if the program validates against unwarranted data entry.	Press 9	The program should provide an input error, and prompt the user to try again.		
5	To see if the program validates against unwarranted data entry.	Press 0	The program should provide an input error, and prompt the user to try again.		
6	To see if the program validates against unwarranted data entry.	Press ttttt	The program should provide an input error, and prompt the user to try again.		
7	To check whether the area of a circle calculates the correct answer.	Radius = 50	Output: 7853.981		
8	To check whether the area of a circle calculates the correct answer.	Radius = 0	As any number multiplied by 0 equals 0, the result should be 0 also.		
9	To check whether the area of a circle validates against negative numbers.	-9	A validation message should appear, informing the user that this is not possible and to try again.		
10	To check whether the area of a circle calculates the correct answer.	Radius = 13.6	Output: 581.068		

Remember that, at this stage, all you need to do is complete the first four columns of the test plan: Test Number, Purpose of Test, Test Data and Expected Results. The actual results and comments will be done later when the programming starts.

Activity 4. You are ready to write and test your program. Use your flow chart, pseudocode and test log to help write and test your program. [30 marks]

You should:

1 write a program that meets the scenario requirements, in a C family or Python® V3.4 (or later version) programming language

2 test your solution to ensure that it functions as expected, and record this testing in your 'Test log and evaluation' document.

Your evidence should include:

1 a copy of your code containing annotations/comments

2 a copy of your test log and evaluation document.

Remember, when you start to write your code, to comment, indent and use correct naming conventions as you develop it. This is essential to attain the highest marks. Notice, within the sample answer code, how all code is separated, indented and appropriate naming conventions are used throughout. Moreover, notice how each section is annotated/commented to explain what each section of coding does. Failure to do this consistently will prevent you from achieving the highest grade.

Your program must be highly efficient and optimised. Therefore split your program into functions or consider using an object-oriented solution to be even more code efficient.

The algorithms that you use in your program to derive the correct calculations must be accurate and output the precise answer. This is essential to achieve the highest grade. Therefore, when undertaking any calculations, remember BIDMAS, as not abiding by this mathematical rule will mean that you derive an incorrect answer.

Notice how the code in the sample answer is also validated throughout. No matter what a user presses, the program will respond with a validation message to inform the user that they have input something incorrectly. This must be done consistently throughout in order to attain the highest grade.

Sample answer: 1. A copy of your code with annotations/comments

```
//libraries
#include <iostream>

//minimise code used within the program
using std::cin;
using std::cout;
using std::endl;

//function and procedure declaration
//parameter passing by reference utilised to keep program modular
void openMainMenu();
```

```
double collectRadius(double &radius);
void calcCircleArea();
void calcCircleCircumference();

double collectLength(double &length);
double collectWidth(double &width);
double collectDepth(double &depth);
void calcSquareArea();
void calcSquarePerimeter();
void calcCuboidVolume();
void calcCuboidSurfaceArea();

void openHelpMenu();
void exitProgram();

int main()
{
   openMainMenu();
   return 0;
}

//menu system to control the flow of the program
void openMainMenu()
{
   int shapeselection = 0;
   cout << "Please choose from the following options:" << endl
      << "Press 1 to calculate a Circle Area" << endl
      << "Press 2 to calculate a Circle Circumference" << endl
      << "Press 3 to calculate a Square Area" << endl
      << "Press 4 to calculate a Square Perimeter" << endl
      << "Press 5 to calculate a Cuboid Volume" << endl
      << "Press 6 to calculate a Cuboid Surface Area" << endl
      << "Press 7 to access the help menu" << endl
      << "Press 8 to exit the program" << endl;
   cin >> shapeselection;

//validation checks. If the cin fails then cin is cleared and ignored
to stop unexpected program behaviour
//if value entered is not one of the menu options then procedure is
recalled after error message
   if (cin.fail())
   {
      cin.clear();
      cin.ignore(INT_MAX, '\n');
      cout << "Input Error. Integer values only" << endl;
      openMainMenu();
   }
   else if (shapeselection < 1 || shapeselection > 8)
   {
      cout << "Input Error. Selection not valid" << endl;
      openMainMenu();
   }
   else
   {
      switch (shapeselection)
      {
         case 1:
            calcCircleArea();
         break;
         case 2:
            calcCircleCircumference();
         break;
```

```
        case 3:
          calcSquareArea();
        break;
        case 4:
          calcSquarePerimeter();
        break;
        case 5:
          calcCuboidVolume();
        break;
        case 6:
          calcCuboidSurfaceArea();
        break;
        case 7:
          openHelpMenu();
        break;
        case 8:
          exitProgram();
        break;
        default:
          cout << "invalid selection" << endl;
          openMainMenu();
        break;
      }
    }
}

void calcCircleArea()
{
    double radius = 0.0;
    collectRadius(radius);
    double area = 3.1416 * radius * radius;
    cout << "Circle area = " << area << endl;
    system("PAUSE");
    openMainMenu();
}

void calcCircleCircumference()
{
    double radius = 0.0;
    collectRadius(radius);
    double circumference = 3.1416 * radius + radius;
    cout << "Circle Circumference = " << circumference << endl;
    system("PAUSE");
    openMainMenu();
}

double collectRadius(double &radius)
{
    cout << "Please enter the radius of the circle" << endl;
    cin >> radius;
    if (cin.fail())
    {
      cin.clear();
      cin.ignore(INT_MAX, '\n');
      cout << "Input Error. Numeric values only" << endl;
      collectRadius(radius);
    }
    else if (radius < 0)
    {
      cout << "Input Error. Negative values are not valid" << endl;
      collectRadius(radius);
    }
```

```
    else
    {
        return radius;
    }
    return radius;
}

//program splits the input collection from the user into separate
functions
//to aid in reusability and modularity
void calcSquareArea()
{
    double length = 0.0;
    collectLength(length);
    double area = length * length;
    cout << "Square area = " << area << endl;
    system("PAUSE");
    openMainMenu();
}

void calcSquarePerimeter()
{
    double length = 0.0;
    collectLength(length);
    double perimeter = length * 4;
    cout << "Square perimeter = " << perimeter << endl;
    system("PAUSE");
    openMainMenu();
}

void calcCuboidVolume()
{
    double length = 0.0;
    double width = 0.0;
    double depth = 0.0;
    collectLength(length);
    collectWidth(width);
    collectDepth(depth);
    double volume = length * width * depth;
    cout << "Cuboid volume = " << volume << endl;
    system("PAUSE");
    openMainMenu();
}

void calcCuboidSurfaceArea()
{
    double length = 0.0;
    double width = 0.0;
    double depth = 0.0;
    collectLength(length);
    collectWidth(width);
    collectDepth(depth);
    double surfacearea = length * width * 2 + length * depth * 2 +
width * depth * 2;
    cout << "Cuboid Surface Area = " << surfacearea << endl;
    system("PAUSE");
    openMainMenu();
}

double collectLength(double &length)
{
    cout << "Please enter the length of the square / cuboid" << endl;
```

```
    cin >> length;
    if (cin.fail())
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout << "Input Error. Numeric values only" << endl;
        collectLength(length);
    }
    else if (length < 0)
    {
        cout << "Input Error. Negative values are not valid" << endl;
        collectLength(length);
    }
    else
    {
        return length;
    }
    return length;
}

double collectWidth(double &width)
{
    cout << "Please enter the width of the cuboid" << endl;
    cin >> width;
    if (cin.fail())
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout << "Input Error. Numeric values only" << endl;
        collectWidth(width);
    }
    else if (width < 0)
    {
        cout << "Input Error. Negative values are not valid" << endl;
        collectWidth(width);
    }
    else
    {
        return width;
    }
    return width;
}

double collectDepth(double &depth)
{
    cout << "Please enter the depth of the cuboid" << endl;
    cin >> depth;
    if (cin.fail())
    {
        cin.clear();
        cin.ignore(INT_MAX, '\n');
        cout << "Input Error. Numeric values only" << endl;
        collectDepth(depth);
    }
    else if (depth < 0)
    {
        cout << "Input Error. Negative values are not valid" << endl;
        collectDepth(depth);
    }
    else
    {
        return depth;
```

```
   }
   return depth;
}

void openHelpMenu()
{
   cout << "This is the help menu" << endl;
   openMainMenu();
}
void exitProgram()
{
   exit(0);
}
```

Sample answer: 2. A copy of your Test log and evaluation document

Test Number	Purpose of Test	Test Data	Expected Results	Actual Results	Comments
1	Output menu information regarding calculator program.	Start program	As soon as program runs a list of options should be presented, asking the user to type in a different number corresponding to the different options.	FAILED Program stated: invalid operands of types 'const char [42]' and '<unresolved overloaded function type>' to binary 'operator<<'	See Screenshots 1a and 1b. To be actioned.
1B	Output menu information regarding calculator program.	Start program	As soon as program runs, a list of options should be presented, asking the user to type in a different number corresponding to the different options.	PASSED Program worked as expected.	See Screenshots 2a and 2b. <Was omitted in cout. Please refer to Screenshot 2a>
2	To see if program will exit when selected.	Press 8	The program should exit when 8 is pressed.	PASSED Program worked as expected.	See Screenshot 3
3	To see if program displays the help menu when selected.	Press 7	The program should present a basic help menu when 7 is pressed.	PASSED Program worked as expected.	See Screenshot 4
4	To see if the program validates against unwarranted data entry.	Press 9	The program should provide an input error, and prompt the user to try again.	FAILED 'openMainmenu' was not declared in this scope.	See Screenshots 5a and 5b. To be actioned.
4B	To see if the program validates against unwarranted data entry.	Press 9	The program should provide an input error, and prompt the user to try again.	PASSED Program worked as expected.	See Screenshot 6. openMainMenu function is case sensitive. Needed capital M for 'Menu'.
5	To see if the program validates against unwarranted data entry.	Press 0	The program should provide an input error, and prompt the user to try again.	PASSED Program worked as expected.	See Screenshot 7

Test Number	Purpose of Test	Test Data	Expected Results	Actual Results	Comments
6	To see if the program validates against unwarranted data entry.	Press ttttt	The program should provide an input error, and prompt the user to try again.	PASSED Program worked as expected.	See Screenshot 8
7	To check whether the area of a circle calculates the correct answer.	Radius = 50	Output: 7853.98	FAILED Program compiled but failed to give the correct answer.	See Screenshots 9a and 9b
7B	To check whether the area of a circle calculates the correct answer.	Radius = 50	Output: 7853.98	FAILED Program compiled but failed to give the correct answer. This time I put brackets around the radius in order to square this first, before multiplying by π. Still not working, will investigate.	See Screenshots 10a and 10b
7C	To check whether the area of a circle calculates the correct answer.	Radius = 50	Output: 7853.98	PASSED Program worked as expected.	See Screenshots 11a and 11b. The previous calculation was incorrect as my initial calculation in testing was pi derived to 11 decimal places. As this was not stated accurately within the program, the answer was wrong.
8	To check whether the area of a circle calculates the correct answer.	Radius = 0	As any number multiplied by 0 equals 0, the result should be 0 also.	PASSED Program worked as expected.	See Screenshot 12
9	To check whether the area of a circle validates against negative numbers.	-9	A validation message should appear, informing the user that this is not possible and to try again.	PASSED Program worked as expected.	See Screenshot 13
10	To check whether the area of a circle calculates the correct answer.	Radius = 13.6	Output: 581.069	PASSED Program worked as expected.	See Screenshot 14

All sections of the test grid must be annotated appropriately and consistently.

Notice how test failures in Test 7 have been repeated more than once until the correct answer was derived. This shows an iterative development process which generates higher marks.

Commenting is clear and concise and links precisely to screenshots, which gives evidence of where the errors were and how they were overcome. This is required for higher marks.

Screenshot evidence for Activity 4

This corresponds to the sixth column in your Test plan, Comments. These screenshots are useful as they demonstrate to anyone who looks at your program that you have identified errors which have occurred within your program and that you have taken steps to resolve them. Screenshots serve as evidence to prove that testing has taken place.

```
//menu system to control the flow of the program
void openMainMenu()
{
    int shapeselection = 0;
    cout < "Please choose from the following options:" << endl
        << "Press 1 to calculate a Circle Area" << endl
        << "Press 2 to calculate a Circle Circumference" << endl
        << "Press 3 to calculate a Square Area" << endl
        << "Press 4 to calculate a Square Perimeter" << endl
        << "Press 5 to calculate a Cuboid Volume" << endl
        << "Press 6 to calculate a Cuboid Surface Area" << endl
        << "Press 7 to access the help menu" << endl
        << "Press 8 to exit the program" << endl;
    cin >> shapeselection;
```

Screenshot 1a

		In function 'void openMainMenu()':
38	56	[Error] invalid operands of types 'const char [42]' and '<unresolved overloaded function type>' to binary 'operator<<'

Screenshot 1b

```
void openMainMenu()
{
    int shapeselection = 0;
    cout << "Please choose from the following options:" << endl
```

Screenshot 2a

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
```

Screenshot 2b

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
8

--------------------------------
Process exited after 2.664 seconds with return value 0
Press any key to continue . . .
```

Screenshot 3

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
7
This is the help menu
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
```

Screenshot 4

```
91          default:
92              cout << "invalid selection" << endl;
93              openMainmenu();
94          break;
```

Screenshot 5a

```
93    18    [Error] 'openMainmenu' was not declared in this scope
```

Screenshot 5b

```
91          default:
92              cout << "invalid selection" << endl;
93              openMainMenu();
94          break;
```

Screenshot 6

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
0
Input Error. Selection not valid
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
```

Screenshot 7

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
ttttt
Input Error. Integer values only
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
```

Screenshot 8

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
1
Please enter the radius of the circle
50
Circle area = 7853.75
Press any key to continue . . .
```

Screenshot 9a

```
void calcCircleArea()
{
    double radius = 0.0;
    collectRadius(radius);
    double area = 3.1415 * radius * radius;
    cout << "Circle area = " << area << endl;
    system("PAUSE");
    openMainMenu();
}
```

Screenshot 9b

```
void calcCircleArea()
{
    double radius = 0.0;
    collectRadius(radius);
    double area = 3.1415 * (radius * radius);
    cout << "Circle area = " << area << endl;
    system("PAUSE");
    openMainMenu();
}
```

Screenshot 10a

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
1
Please enter the radius of the circle
50
Circle area = 7853.75
Press any key to continue . . .
```

Screenshot 10b

```
void calcCircleArea()
{
    double radius = 0.0;
    collectRadius(radius);
    double area = 3.14159265359 * (radius * radius);
    cout << "Circle area = " << area << endl;
    system("PAUSE");
    openMainMenu();
}
```

Screenshot 11a

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
1
Please enter the radius of the circle
50
Circle area = 7853.98
Press any key to continue . . .
```

Screenshot 11b

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
1
Please enter the radius of the circle
0
Circle area = 0
Press any key to continue . . .
```

Screenshot 12

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
1
Please enter the radius of the circle
-9
Input Error. Negative values are not valid
Please enter the radius of the circle
```

Screenshot 13

```
Please choose from the following options:
Press 1 to calculate a Circle Area
Press 2 to calculate a Circle Circumference
Press 3 to calculate a Square Area
Press 4 to calculate a Square Perimeter
Press 5 to calculate a Cuboid Volume
Press 6 to calculate a Cuboid Surface Area
Press 7 to access the help menu
Press 8 to exit the program
1
Please enter the radius of the circle
13.6
Circle area = 581.069
Press any key to continue . . .
```

Screenshot 14

Activity 5 Evaluate your program solution. [12]

You should cover:

1 How well your solution meets the requirements of the scenario.

2 The quality and performance of your program.

3 The choices you made about coding conventions.

4 The changes you made during the development process.

For a question that uses the word 'evaluate', you must look at the whole approach you took, and be sure to include what went well and what did not go so well.

In order to gain the higher marks, you must ensure that you make valid and fully supported judgements throughout your evaluation. 'Mostly' supported arguments will not achieve the highest grade.

Make certain that you make logical coherent links between the solution and the requirements set out within the scenario throughout your evaluation.

Sample answer:

The solution I developed was implemented using C++. I used this programming language because I have had lots of experience using it, compared with using Python. I began by analysing the scenario and determining what was required. I did this by making notes on what the functionality of the program should be, in other words, what the program will do. Once I knew this, I began to design my solution.

I used flow charts and pseudocode to develop my solution. Before developing my code program solution, I knew it was vitally important that my designs were correct. If I developed a design which was incomplete or incorrect, then

this would mean my solution would follow suit. Therefore, I took enough time to make certain that my design techniques included all of the functionality set out in the scenario. Once I was happy with my design documentation, I then proceeded to make my programming solution.

When developing the design solution, I was aware that the scenario requested that I create a program that would perform different calculations based on attributes of shapes. I knew it was critical in order to give accurate results that I implemented formulae to give the correct algorithm. I was especially conscious of BIDMAS (Brackets, Indices, Division, Multiplication, Addition, Subtraction) rules when applying formulae. For example, when writing code for the cuboid surface area I realised that the code would perform the multiplication elements first, before adding them up, so that I did not need to add brackets. I knew that within the scenario I needed to prove 'my worth' as it stated that they needed to restrict candidate numbers to call for interview. Therefore, if I had made a mistake such as this, the decision may well have gone against me. This is a good practice that I can take with me when I apply for software development positions in the future.

I devised suitable test results to ensure that my program output the correct answer. I used a testing strategy called 'black box testing'. Essentially, I developed a series of tests prior to my creating the code for the solution. Therefore I knew what the correct answer should be in each case when I input figures for the different shapes. Having created the program and tested it, I can now say that my programming solution meets all the requirements for determining the correct answer for each shape. In each instance in my program, if you put figures into any shape it will determine the right answer. My testing confirms this, which tells me with enough certainty that the program is right when outputting the result.

I also needed to make certain that my program validates: in other words that the program will deal with any unexpected input. For example, when determining the area of a square the formula is length multiplied by width, which means you need to input numbers into the program. However, what would happen if a user input text instead? Therefore, I ensured that this (and every other occurrence of data entry) is validated so that it prompts the user again if there is unexpected or erroneous data entry. A good example of this is in the menu system where it ensures that only numbers between 1 and 8 can by typed. If this condition has been met, then a switch statement will determine what number has been pressed, and will return the correct function based on user input. However, if any other number (not 1 to 8) is input, then this will output an error message based on an if statement. I used validation techniques such as this one throughout to ensure that my program is properly robust and will deal with any unexpected data entry.

In summary, I can conclude that the program I have developed meets all of the requirements of the scenario which had been set out. I can confirm through the testing process that the program not only derives the correct answer for all shapes, but also validates against unexpected data entry.

I was conscious when developing the programming solution that it had to perform well. I knew that one of the key features of good design is modularity. Modularity is where independent components are developed in isolation and can then be integrated to form a complete computer program. I wanted to ensure that my solution made use of this feature, as it would enable me to complete different parts of the program, test them and then move onto the next part independently of the other parts. The way in which I did this was to develop various functions throughout my program. I developed my own functions for each part of the program. For example, I developed a function called 'calcSquareArea' which was responsible for taking in the width and then determining the area of the square. As this has been written into a function, it can be reused and it also makes it easier to test and compile the programming solution.

I also wanted to ensure that the quality of my program was a good as it could be. The way in which I achieved this was to check that the program derived the correct answer. I ensured that there was thorough testing done prior to programming. I derived correct test data so that I knew what the right answer would be for any given shape. Therefore, when I tested my program solution, I knew that when an actual result was given I could compare this against my expected results. There were instances when they did not match so I went back to the code and made amendments to it. I then performed remedial testing on the code, and in the end the answer derived by the program (the actual results) and the test data (expected results) matched. Therefore I knew that the program was correct and was of good quality.

One of the key details I wanted to make sure my program had was correct programming conventions throughout. I understood that these conventions are universal and should always be applied when coding a solution. One aspect of this was to ensure that my program is commented on throughout. In my code, I made sure that commenting was done through the use of '//' or '/'. These conventions are used so that another person could look at my code and understand clearly what it is responsible for doing. Moreover, it also demonstrates to anyone looking at the code that I understand what it is doing. In the scenario provided, if I am successfully shortlisted for interview and then get the job, I would be working within a team of programmers. Commenting on my code shows that I am using this good practice and that other members of the team would be able to understand the code I have developed.*

Another coding convention I have used is to indent my code. A reason for using C++ is that it automatically indents my code. This made it easier to program, because when I used a brace bracket '{}' it automatically indented it to where it needed to be. I know that by indenting code it makes it easier to read. I also ensured that my program had correct naming conventions throughout. For example, if you were to look at my code I used proper variable names such as 'area', 'length', 'width' and so on. These terms would make sense to anyone who looked at my program and make the code more intuitive. If I had used naming conventions such as 'num1', 'num2'

and 'num3', then it would make the program more difficult for another programmer to decipher.

In summary, I am very happy with the work that I have produced. By having a good design and a strong test plan in place, this ultimately led me to develop a robust and understandable program. I am confident that, with the program I have developed, I would be shortlisted for interview as per the scenario.

IT Systems Security and Encryption 7

Getting to know your unit

Assessment
You will be assessed by a series of assignments set by your tutor.

Security is a big issue in today's internet-centric world. We rely heavily on computer systems for so many aspects of our lives that they have become an attractive and lucrative target for the modern criminal. Keeping computer systems secure requires constant vigilance because, as technology develops rapidly, new threats to these systems, and new vulnerabilities, appear all the time.

As a computing professional, you will need to have a good understanding of current security threats and of how to apply protection methods for any given cyberattack. As security defences become more robust, the attack methods become more sophisticated.

In this unit, you will learn about the different types of security attack and how to protect IT networks from these attacks, as well as how to configure and support these networks. You will learn about encryption and will apply suitable protection to an IT system. A good understanding of cybersecurity will prepare you for a job in the computing industry or for entry into higher education.

How you will be assessed

This unit will be assessed by a series of internally assessed tasks set by your tutor. Throughout this unit, you will find assessment activities that will help you to prepare for the final assessment.

Assessment criteria

This table shows what you must do in order to achieve a **Pass**, **Merit** or **Distinction** grade, and where you can find activities to help you.

Pass	Merit	Distinction
Learning aim A Understand current IT security threats, information security and the legal requirements affecting the security of IT systems		
A.P1 Explain the different security threats that can affect the IT systems of organisations. **Assessment practice 7.1**	**A.M1** Assess the impact that IT security threats can have on organisations' IT systems and business while taking account of the principles of information security and legal requirements. **Assessment practice 7.1**	
A.P2 Explain the principles of information security when protecting the IT systems of organisations. **Assessment practice 7.1**		
A.P3 Explain why organisations must adhere to legal requirements when considering IT systems security. **Assessment practice 7.1**		**AB.D1** Evaluate the effectiveness of the techniques used to protect organisations from security threats, while taking account of the principles of information security and legal requirements. **Assessment practice 7.1**
Learning aim B Investigate cryptographic techniques and processes used to protect data		
B.P4 Explain the principles and uses of cryptography to secure and protect data. **Assessment practice 7.1**	**B.M2** Analyse how the principles and uses of cryptography have an impact on the security and protection of data. **Assessment practice 7.1**	
Learning aim C Examine the techniques used to protect an IT system from security threats		
C.P5 Explain how protection techniques can help defend an organisation from security threats. **Assessment practice 7.2**	**C.M3** Justify the choice of protection techniques used to defend the IT systems of an organisation, showing how its IT system will be protected from security threats. **Assessment practice 7.2**	**CD.D2** Evaluate the plan and the effectiveness of the protected IT system against requirements. **Assessment practice 7.2**
C.P6 Produce a plan to protect an IT system that meets organisational and legislative requirements. **Assessment practice 7.2**		
Learning aim D Implement strategies to protect an IT system from security threats		
D.P7 Perform tasks to protect the IT system to meet requirements given in the plan. **Assessment practice 7.2**	**D.M4** Enhance the protection of the IT system to meet requirements given in the plan. **Assessment practice 7.2**	**CD.D3** Demonstrate individual responsibility and effective self-management in the planning and protection of an IT system. **Assessment practice 7.2**
D.P8 Review the extent to which the organisation's IT system has been protected. **Assessment practice 7.2**		

Getting started

Have you or someone you know had a security issue with a computer or suffered from a computer virus? These kinds of problem are very common, but they can cause a lot of disruption. How much time did you or the person you know spend fixing the issue?

A Understand current IT security threats, information security and the legal requirements affecting the security of IT systems

Link

Cybersecurity issues are often in the news. Check out news websites such as the technology section of the BBC news website for the latest IT security issues.

Resolving security issues costs not only time but also money, if you need to pay to have the virus removed. Security issues may be very frustrating too, because you may lose important files. The frustration and annoyance that individual users encounter when they have security issues with a personal computer system is amplified many times for businesses as they stand to lose significant amounts of money, both directly (due to fraudulent activity) and indirectly (through the time required to resolve security issues). Businesses often want to keep quiet about any security issues they encounter as this may reflect badly on the organisation, so the true extent of the security problems experienced by businesses is not easy to confirm. If you ask your friends and family, you may well find that almost everyone who uses a computer (personally and professionally) has experienced some kind of security issue, even if only by receiving a phishing email. Cybersecurity is never far from the news.

Threat types

Threats to IT systems can come from a variety of sources. A company's employees, either deliberately or unintentionally, can be a source of security risk, while a range of threats also exist outside the company. The threats themselves come in a number of different forms: internal, external, physical or social engineering and software-driven threats.

Internal threats

There are many threats to IT security from within an organisation, both deliberate and accidental or unintentional.

Deliberate

There are a number of deliberate, internal threats to security.

- **Employee actions:** Disgruntled or recently dismissed staff may damage or delete files as a form of revenge. They may also cause damage to the company software systems or post damaging information about the company online.
- **Data theft:** Employees may steal data (such as customer lists or credit card details) which they can then sell to cybercriminals or competitors. In many cases, it is relatively easy for employees who have legitimate access to an organisation's computer systems and data, to make copies of it.
- **Users overriding security controls:** Employees may want to bypass security controls as they find them too restrictive and frustrating. For example, they may want to use their favourite internet browser or install a game they enjoy playing during their lunch break.

Tip

As a computing professional you may find yourself working with users who have a limited understanding of IT security issues. You need to develop the communications skills to be able to explain in a clear and non-technical way what the security issues are and why protection methods are in place. You also need to remain calm and professional when users may be angry or upset because of the IT problems they are facing.

Accidental or unintentional

There are a number of ways in which internal security can be threatened accidentally or unintentionally.

- **Accidental loss:** Employees may accidentally delete important data. Accidental deletion or corruption of data may be due to poor training or misunderstanding of procedures, or it could be due to poor software application design. Although most operating systems provide a 'recycle bin' facility where deleted files may be retrieved, some of the software applications an organisation uses may not have this facility.
- **Unintentional disclosure or damage to data:** Employees may unintentionally disclose confidential data (such as passwords) or they may damage data by corrupting it. Employees may unintentionally disclose their passwords by writing them down (perhaps even leaving notes on their desk with passwords written on them), by leaving their computer logged on when they are not at their desk, or by sharing passwords with others. By disclosing passwords to anyone who does not have legitimate access to the computer system, an employee is exposing the organisation to a security risk because the information may fall into the hands of a cybercriminal. Corruption of data may be due to poor training or misunderstanding of procedures, or it could be due to poor software application design, facilitating accidental deletion of data.
- **Unsafe practices:** An organisation's employees may carry out actions which leave the company's computer system open to security attacks by external cybercriminals. These unsafe practices include the following.
  - Using external flash (USB) storage devices which may have been infected with **malware** when used on another system, such as the employee's home computer.
  - Visiting websites which are untrusted, because these websites may attempt to infect computer systems with malware.
  - Downloading files from untrusted websites may introduce malware to a computer system. Uploading company files to the internet also exposes company data to cybercriminals.
  - Using file sharing software applications may introduce malware to a computer system.
  - When employees bring in their own devices (laptops, tablets, smartphones) from home (sometimes called 'Bring your own device (BYOD)') and connect them to the organisation's network, it exposes the network to any malware that is on those devices. Some companies encourage employees to use their own devices as then the company does not need to provide them and it can boost morale, but they need to protect themselves from the related threats to security by addressing BYOD in their acceptable use policy.

Link

For more about acceptable use policies, see 'Policies and procedures'.

Key term

Malware – an umbrella term for software that has a malicious intent. Malware has a deliberately negative effect on a computer. Once installed, it can gather private information, slow the computer down, delete or lock access to files, for example.

External threats

Threats from outside an organisation come from a variety of sources. In most cases, there is a financial motivation and cybercriminals usually intend either to steal information which can be used to obtain money (that is, bank account details or credit card numbers) or to hold a company or individual to ransom by encrypting data or preventing access to services. Competitors may also wish to attack an organisation in order to gain a competitive financial advantage. Such activities would most likely be themselves criminal acts.

In some cases, there is a political motive, for example demonstrators or protest groups may target an organisation that they disagree with politically, ethically or on the basis of religion. Cyberattacks may also be used as a form of warfare when one country attacks another. There are many benefits to cyberterrorism for the attacker, not least of which is that there is little risk of loss of life for the attacker, but the chaos that would be caused if banking, transportation, or other financial or safety critical systems were disrupted or destroyed by an attack would be huge.

Research

The Stuxnet computer worm is said to have been developed by the US and Israeli governments to attack the Iranian nuclear programme, although they have never admitted this. Research what Stuxnet does and how it has been used.

Physical threats

Computer equipment can be valuable and there is a possibility that it may be stolen or maliciously damaged. Theft or loss of portable equipment which may contain sensitive information is a particular threat to security because once a hard disc is removed from a computer system, it may be possible to circumvent the security measures that protect the data it holds. Organisations also need to be aware of the possibility of accidental damage or destruction to computer systems by fire, flood or other disasters. This may mean the loss of important and valuable data and, at the very least, would result in the expense of replacing the equipment.

Link

There have been a number of occasions when people have lost laptops containing personal or sensitive information. A search on the BBC news website should produce several news articles about laptops containing sensitive or personal data which have been lost or stolen, and the potential consequences.

Social engineering and software-driven threats

Social engineering is a technique that is used to attempt to fool computer users to provide secure information to cybercriminals. Social engineering uses software that is designed with a malicious intent.

Social engineering

The best known example of social engineering is the 'phishing' scam. This involves sending emails that claim to come from a bank to large numbers of people asking them to log into their account using a link provided in the email. (Quite often the people receiving the phish email will not even have an account with the bank that the email claims to come from). The link provided in the email does not take the receiver to the real bank's website but to a fake version of the site where, if the individual enters their banking login credentials, these can be stolen by the cybercriminals running the scam.

There are many other scams which use social engineering, and some are as simple as telephoning someone in an office claiming to be from their IT support department and asking them for their passwords. Social engineering threats can be difficult to defend against and there is no software protection available, so people need to be trained to be aware of these kinds of threat and to be on their guard at all times.

Some of the well-known social engineering threats with specific names include the following.

- **Shoulder surfing** – This is simply looking over someone's shoulder in an attempt to obtain sensitive information such as usernames, passwords and PINs. Careful placement of monitors can help prevent this.
- **Spear phishing and whaling** – These are both forms of phishing that involve targeting a specific group of people rather than the blanket approach used with phishing. For example, a criminal might send an email to all the employees in a company that claims to come from one of the company directors, which asks them to reply to the email with their system password. This is known as spear phishing because it is targeted phishing. Because the email looks as if it comes from the company director, employees might be fooled into responding. Whaling involves sending phish emails to senior executives ('whales') specifically.
- **Dumpster diving** – Looking through rubbish or recycling bins (dumpsters) can provide criminals with valuable information. They might, for example, be able to obtain a company employee directory that contains the emails and phone numbers of everyone in the company. This information could then be used for a spear phishing or whaling attack.

Discussion

Discuss with a peer or in a small group the ways in which you can defend yourself and others against social engineering attacks such as phishing.

Software-driven threats (malware)

Social engineering cyberattacks use malicious software, known as malware. Malware is one of the best known threats to IT security and, although the first malware was originally created in the 1980s as an experiment or prank, malware is now a serious threat to computer systems.

Malware is very common and large amounts of malware are released every year. The main route for malware to reach computers is via the internet, although infection via removable devices such as USB drives is also common.

Malware usually targets Microsoft® Windows® computers, and is much less prevalent on other operating systems such as Linux®. The increasing use of mobile devices has led to more and more malware being created to attack Android™ and Apple® iOS.

There are a number of different types of malware.

- Viruses are programs that are usually concealed within another program or file. They replicate by inserting copies of themselves into other programs or files. They usually (but not always) have a malicious intent.
- Worms, like viruses, also replicate themselves, often over a computer network, but, unlike viruses, they do not attach to another file or program. Worms often seek out known security flaws and 'worm' their way into the system through these 'holes' – hence the name.
- Trojan horse is a term for any type of malicious program that pretends to be something useful or interesting in an attempt to get a user to unwittingly download or install it.
- Ransomware is a type of malware that restricts access to a user's computer, often by encrypting files, and demands a ransom be paid before the computer will be unlocked.

Research

CryptoLocker is a well-known example of ransomware which encrypted files on infected computers and demanded a ransom for their decryption.

Research how CryptoLocker works, what effects it had on infected computers and how it was eventually disrupted.

- Spyware collects information without the user's consent and is most commonly used to collect information about a user's internet browsing habits with the aim of showing targeted pop-up adverts to the user. Some types of spyware include an embedded 'keylogger'. This records any information you type at your keyboard which can be sent on to someone else who may have a malicious intent
- Adware is malware that presents adverts to the user, usually using pop-up windows. It may (like spyware) analyse a user's internet habits to provide targeted adverts. Spyware and adware do not normally have a damaging effect on a computer but can be irritating for the user.
- Rootkit is term for a program which can allow an attacker access to areas of a computer that they would not normally have access to. For example, it might provide the attacker with administrator or root access to the operating system.
- Backdoor is a method that is used to bypass a computer's normal authentication procedures, thereby providing unauthorised remote access to the computer.
- Logic bomb is code included in malware that lurks for a while and then executes when certain conditions are met, such as a specific date being reached. Such malware can spread widely across computer systems before it is noticed.

Computer network-based threats

Many security threats access the targeted computer system via the networks they are connected to. Many of these threats use features of the **internet protocol (IP)**.These computer network-based threats can be passive or active.

Key term

Internet protocol (IP) – one of the most important protocols is the TCP/IP protocol suite which is used for communication in local area networks (LANs) and via the internet. IP addresses are assigned to all devices that are participating in a network that uses the internet protocol.

Passive threats

Passive computer network-based threats include the following.

- Wiretapping: This pre-dates computers, having been used since the early days of telephone networks, and it involves listening in to data being transmitted over a network. Traditionally, this involved a physical connection to a wired network but, more recently, it is typically done by listening into WiFi networks that use radio communication. WiFi networks should be encrypted so that only users who know the key to access the network can join it, but it may be possible to crack a WiFi key or obtain it through social engineering techniques. An attack may use techniques such as **ARP spoofing** and software such as Wireshark to view data sent over the network.

Key terms

ARP spoofing – ARP (address resolution protocol) is a protocol used in LANs to provide a mapping between the physical addresses of devices (**MAC addresses**) and their IP addresses. An attacker can send fake (spoof) ARP messages onto a local area network (LAN) which associate the attacker's computer with the address of another device on the LAN, such as the default gateway (or router). Therefore all messages being sent outside the network will be forwarded to the attacker's computer rather than to the default gateway.

MAC address – a unique fixed hardware address that is permanently coded into a network device, such as a network interface card, by its manufacturer. A MAC address is a 12 digit hexadecimal number. The allocation of MAC addresses is managed by the IEEE (Institute of Electrical and Electronic Engineers).

- Port scanning: This is a method of probing a computer's network **ports** to see if any of them are open and what services are running on the system. This is not an attack, as such, but a would-be attacker can use this information to potentially exploit vulnerabilities in the services that are running.

Key term

Ports – a part of the design of the TCP/IP protocol suite that is the method of communication that the internet uses. IP addresses are used to identify individual computers, and ports are used to identify specific services or applications. The HTTP protocol, used to transfer web pages, generally uses port 80.

- Idle scanning: This is a form of port scanning in which the attacking computer does not have any direct interaction with the computer it is scanning. The attacking computer uses another computer called a **Zombie** to scan the target computer's ports. The benefit to the attacker of using this method is that there is no trace of the attacking computer's address on the target machine.

Key term

Zombie – a computer which has been compromised by an attacker and used to carry out a variety of malicious attacks on other computers under the remote control of the attacker. The owner of the computer is unaware that this has happened.

Active threats

There are also many computer network-based threats which can be classified as active threats.

- Denial of service (DoS) attack: Involves sending so many requests to a web server that the server is overwhelmed and cannot respond to legitimate requests. This has the effect of taking the website hosted on the target machine off-line. In most cases, the sender IP address of the machine launching the attack is forged so the source of the attack cannot be easily identified. There are a variety of different versions of the DoS attack. In order to send sufficient requests to overwhelm a powerful web server, a distributed denial of service (DDoS) attack can be used, where a **botnet** of many computers can be used to launch the attack.

Key term

Botnet – a collection of zombie computers (very often home-based computers) which are controlled remotely to carry out various actions, such as sending spam emails or launching DDoS attacks.

- Spoofing: This is a technique used in many types of attack (such as DoS attacks) in which the source address of an attacking computer is hidden. For example, it is possible to create IP data packets with a forged sender IP address. This helps to hide the identity of the computer launching the attack.
- **Man-in-the-middle:** This type of attack involves listening in to, and potentially modifying, data exchanges between two computers. Man-in-the-middle (MITM) is a type of eavesdropping and could be used, for example, on an unencrypted WiFi connection to view data being sent by people using the network.

- **Address resolution protocol (ARP) poisoning:** The ARP is used to map fixed physical hardware addresses (MAC addresses) to IP addresses (which are dynamically allocated to devices by routers or servers) on a LAN. In this type of attack, faked ARP messages are sent which are intended to fool other devices that the attacker's computer is actually another device, such as the network default gateway (where messages which have destinations outside the LAN are sent). This allows the attacker to intercept messages that were not intended for their computer. This is a type of MITM attack.
- **Smurf attack:** This is a type of DDoS attack. It utilises a feature of the internet control message protocol (ICMP), whereby, when a computer receives an ICMP message, it replies to the source IP address in the ICMP message, confirming that the message was received. The original intention of this feature was to test if devices were connected and working. However, an attacker can send an ICMP message with a faked source IP address of the computer to be attacked. If this message is sent to the broadcast address on a network, then every device on that network will reply, sending its message to the computer that is being attacked (the broadcast address is a special IP address. Messages addressed to it are sent to every host on the local network). In a large network, so many ICMP messages could be sent that the attacked computer would be overwhelmed. To avoid this kind of attack, most modern devices no longer respond to ICMP messages sent to the broadcast address and so are not vulnerable to this type of attack.

Research

ICMP is not used to send messages between computers. Instead it is used for diagnostic or control purposes. One useful network diagnostic tool that uses ICMP is a program called Ping. Find out what Ping is for and how it can be used.

- Buffer overflow: This is an example of a vulnerability that can affect legitimate programs running on a system which can be of benefit to an attacker. A buffer is a memory area allocated by a program to store input data. The program will allocate a certain amount of memory but, if the input is larger than the buffer, an overflow can occur. There are a number of ways in which this can be exploited by cyberattackers and they all require an advanced understand of programming techniques. Essentially, this involves forcing the program to execute code provided by the attacker rather than the intended code.
- Heap overflow: This is a type of buffer overflow. In operating system memory management systems, the 'heap' is a term used to refer to those areas of memory that can be allocated to programs for them to use. By corrupting application data held in the heap in a particular way, the attacker's code can be executed.
- Format string attack: This is another type of attack that can target poorly written programs that use the C family of programming languages. The attack targets code used to format input data. Carefully designed input causes this code to behave in a way that can be of benefit to the attacker. For example, it might allow the attacker to execute their own code. A detailed understanding of programming is required to exploit this vulnerability.
- Structured query language (SQL) injection: SQL injection is a widely used method of attack that targets SQL databases and is often used against web-based database applications. This attack involves using certain types of input that can change the way the application works via a web page. It provides the attacker with the ability to access data in the database in a way that the application did not intend.

Worked Example: Product search SQL injection

For example, think of the product search box that appears on e-commerce websites. When you type something into the search box (such as 'Nike® trainers') that value is used in an SQL query that searches website's database for matching values. Typically the sort of SQL statement that might be used is as follows.

SELECT * FROM Products WHERE description = 'Nike® trainers'

An attacker with a knowledge of SQL can use the fact that SQL uses a semicolon to indicate the end of a statement. Suppose that an attacker entered the following into the product search box:

'Nike® trainers'; SELECT * FROM customers;

When this statement is read by the web application, it is turned into two statements (because the semi colon indicates the end of the first one). The first one is the same as before, the second one is:

SELECT * FROM customers

This could potentially provide a list of all the records on the customer table, including names, addresses, credit card numbers etc. []

There are a number of other similar ways that search string inputs can be used to manipulate SQL statements. By sending improperly formatted SQL statements that generate errors, the error messages shown can tell the attacker a lot about the software the web server is using, such as the type and version of database system in use. This information is useful in that it helps the attacker to exploit vulnerabilities which exist in particular software and versions of that software.

SQL injection can allow attackers access to large amounts of data which may be confidential, for example customer details. Applications should be written in such a way as to prevent this kind of attack using proper input validation and error checking, but many older applications are not protected in this way and are vulnerable to attack.

Research

SQL injection has been used in a number of high profile attacks on well-known British companies in which large quantities of customer details have been accessed. Research SQL injection attacks and find out what happened in some well-known cases and find out why the systems were not protected.

- **Cyberattack:** An attack against a specific organisation or government using any of the methods described above. Cyberwarfare is becoming increasingly common and many governments have prepared themselves to defend against these attacks.

Cloud computing security risks

The recent trend in cloud computing brings additional security risks. Cloud computing involves the use of internet-based data storage and computing resources. Rather than using computer resources located within an organisation, the organisation accesses shared resources which are owned and run by a third-party data centre. This enables organisations to react faster and be more flexible, with the ability to fairly rapidly increase their computing resources when demand is high and also reduce them when demand drops. This is something that cannot easily be achieved with in-house computing resources, which usually involve investing in hardware. Organisations may also set up their own cloud systems to provide more flexible access to in house applications and data.

Cloud computing has its own security issues. Some argue that using cloud computing may improve the security of data because a large third-party data centre is able to devote greater resources to security than an individual company can. However, because these large data centres hold large amounts of data from many different clients on their systems, they become attractive targets for cybercriminals.

Organisations that store their data on the cloud no longer have physical access to the servers and disc on which their sensitive data is stored. This data is instead exposed to internal attacks from within the third-party company that hosts the data. The hosting company must take extra precautions to protect against internal attacks. The data held for one organisation by a cloud computing hosting company may also share the servers and disc drives with data held for other organisations, so the hosting company must ensure that each client's data is properly isolated from other clients'.

The server computers in data centres are often virtualised (known as **server virtualisation**), with one physical computer running a number of different virtual machines for clients. The virtualisation software adds an additional layer of complexity to the system and creates the possibility that vulnerabilities may exist in the virtualisation software.

Key term

Server virtualisation – it is very common for server computers to use virtualisation software that allows one large and powerful server computer to operate as if it were several smaller systems carrying out specific functions. In cloud computing, this allows a data centre to provide a customer with what appears to be their own dedicated server computer. In reality, it is just one of many different virtualised servers running on a single large physical server.

Research

Carry out research to find out what the most common threats are to computer systems. There are a number of places you can go to research this question. The Open Web Application Security Project (OWASP) publishes a list of the 'top 10' web security risks from time to time. Companies that produce anti-malware software, such as Norton™, McAfee and Kaspersky™, also publish lists of the most destructive malware

Information security

Much of our personal and business information is now held on computers, so keeping information secure is an extremely important concept. Information needs to be available to people who should have access to it, but protected from those who should not. There are three main principles that apply to information security: confidentiality, integrity and availability.

Confidentiality

Confidentiality is the principle that information should not be disclosed or accessible to anyone who is not authorised to know it. Usually, confidentiality of data on computers is achieved using encryption, which makes it unreadable to anyone who is not authorised to access it. Operating systems have access controls that allow system users to be identified (by their username), authenticated (by their password) and authorised (by the file access permissions granted by the system administrator) so that information held on the system is only accessible to those who should be able to see it.

Integrity

Integrity of data on computers means that the information is accurate and complete, and also that it is not possible for unauthorised changes to be made to the information. Hashing provides a way to check the integrity of digital files. A 'hash' is simply a number produced by applying a hashing algorithm to the file.

Link

Hashing is explained in more detail in the section 'Cryptography methods'.

Digital signatures and certificates also provide integrity, by providing assurance that an email is from the person from whom it appears to be. In database systems the normalisation process used at the design stage is used to help ensure data is not duplicated which can impact on integrity. Bear in mind also that out of date or inaccurate information can impact on the integrity of a database.

Link

Digital signatures and certificates are explained in more detail in the section 'Digital certificates', in 'Applications of cryptography'.

Digital signatures can also provide **non-repudiation**, as do operating system features such as audit logs, which show which particular user accessed a file and when.

Key term

Non-repudiation – this is the assurance that something cannot be denied (repudiated). Typically, this involves using a digital signature to prove that a contract or email was agreed to or sent by someone and they cannot deny it.

Availability

For information to be useful it must be available to those people and systems who need to have access to it. Therefore system managers must take steps to ensure that information systems provide the required level of availability to users. An organisation must decide what level of availability it needs for its computer systems. For some companies, Monday to Friday 9 am to 6 pm is sufficient; for others there must be 24 hour, 365 day availability of systems.

An organisation must also decide how long it can survive without its systems in the case of hardware failure or a disaster such as a fire or flood. In order to increase the availability of a system, an organisation has to remove any single point of failure from the system. They can achieve this through a number of methods, including the following.

- **Disc redundancy** – using multiple discs systems, such as RAID, to allow the system to continue uninterrupted if one disc fails.

Research

RAID disc systems are commonly used on server systems and often include 'hot swap' disc drives which can be removed while the system is running. Research RAID systems and associated hardware features, and consider what security implications they may have and how they can be mitigated.

- Backup – to replace any data lost due, for example, to corruption or human error.
- Server redundancy – duplicated hardware in case of a serious hardware failure.
- Disaster recovery – putting plans in place to run the whole system from a different location in the case of a physical disaster such as fire or flood.

Link

System availability is discussed in more detail in the section 'Physical security'.

Access

Companies need to haveconsidered what level of access can be given to specific users or groups of users, particularly with regard to systems that contain a lot of sensitive data. The ideal situation would be that each user only has access to the data that they actually require for doing their job. This is the principle of providing minimal access to information or lowest required access permission, so as to maximise protection of data.

Most systems allow various levels of access, which are usually defined as:

- full access – ability to read, write and delete files
- write access – ability to read and write files but not delete them
- read access – ability only to read (view) files.

In a large organisation with many employees, it may not be possible to define exactly which files and what level of access each individual user should have in order merely to do their job, so user access rights to files are normally allocated to groups of users, perhaps based on the department they work for or their seniority within the company.

As well as giving the correct level of access to the users who should be able to access the data, the system should protect the data from unauthorised access or modification of information and from theft, because personal, financial and commercially sensitive information (or intellectual property) may be valuable to others. The system should also protect data from deliberate or accidental loss, though the use of backups.

Legal requirements

Legislation regarding security risks can broadly be divided into those laws which organisations must comply with and those under which people who attack systems can be prosecuted.

Must comply with

In the UK, organisations must comply with the following legislation to stay within the law.

Data Protection Act (1998)

This legislation defines the requirements for organisations which legitimately store personal information about living individuals. Any organisation that stores data about living individuals must make sure that the data is:

- used fairly and lawfully
- used for limited, specifically stated purposes
- used in a way that is adequate, relevant and not excessive
- accurate
- kept for no longer than is absolutely necessary
- handled according to people's data protection rights
- kept safe and secure
- not transferred outside the European Economic Area without adequate protection.

Link

For more about the Data Protection Act (1998), visit the Information Commissioner's website: https://ico.org.uk/

Copyright, Designs and Patents Act (1988)

Copyright is the law that protects the intellectual property of writers, musicians and other artists. People who illegally copy music, films, books or computer games are breaking this law. Various methods can be used to help protect intellectual property, such as encryption.

Telecommunications (Lawful Business Practice) (Interception of Communications) Regulations (2000)

Employers are, under these regulations, allowed to intercept communications sent over their own networks. So, for example, a company could lawfully intercept their own employee's emails and record their telephone conversations (if they use the company network). Employees must be aware that their communications may be intercepted, and this would usually be included in their contracts of employment.

Criminal acts

The following legislation outlines criminal acts in relation to computers and data.

Computer Misuse Act (1990)

This is the main legislation that applies to computer hacking and the creation and dissemination of malware. The Act, which was passed in 1990, makes it an offence to:

- make unauthorised access to computer data
- make unauthorised access to a computer system with the intention of committing a crime
- make unauthorised modifications to computer data
- create, supply or obtain anything which can be used to do any of the above.

Fraud Act (2006)

This is a wide-ranging law which covers a variety of fraudulent activities. Fraud is basically a deliberate attempt to deceive someone and obtain some kind of advantage by doing so (often, but not always, monetary). Phishing scams, for example are a form of fraud, since the person sending the emails and setting up the fake bank website is falsely claiming to represent a bank. This is sometimes

called 'fraud by false representation'. However, bear in mind that cybercriminals may not be based in the UK or EU and, therefore, obtaining a conviction under this law for phishing could be very difficult indeed.

Legal liability and contractual obligations

Companies that supply a product or service to customers have contractual obligations to them and can also be legally liable. These companies must set out contracts for their services which state the details of the services they will supply and the associated conditions of those services. For example, a company that provides IT support will have a contract with the business to whom it supplies the support. In an IT support contract, there are likely to be conditions, such as keeping the personal data of customers safe. If a security breach led to the personal data of the IT support company's customers being stolen, then the IT support company would be in breach of their contract and could be sued. If a company suffers a security breach that causes financial or other losses to other people, then they may be legally liable and may have to pay damages.

Impact of security breaches

Security breaches can have a serious impact on organisations or individuals. A serious security breach is likely to result in one or more of the following consequences.

Operational impact on an organisation of the loss of data or service

A computer system on which an organisation relies to run some aspect of its business may be subject to a security breach that prevents the system providing the service it is designed to provide (such as during a DoS attack). This can have a serious impact on the organisation's ability to do business. Attacks that damage, destroy or restrict access to data held on an organisation's computer system would have a similar effect. Many companies rely very heavily on computer systems to run their businesses and so these kinds of security breach can have a devastating impact on them.

Financial impact of loss of service, such as an e-commerce website

Many organisations rely on computer systems to directly sell their products or services, such as those which run e-commerce websites. If an e-commerce website is off-line due to a security breach, a business will lose revenue because customers cannot purchase at this time and they could lose very large sums of money for every hour that the site is off-line.

Damage to reputation

For large and well-known organisations, the impact of a major security breach can spread beyond the financial effects. Many organisations, especially those that do business on the internet, have a reputation to protect. People who buy their products or services online need to feel that the financial and personal information they give to the business is secure. If it becomes public knowledge that a business has suffered a security breach, then this could damage their reputation and lose them customers because they are no longer trusted.

Legal consequences of data privacy breaches

As mentioned earlier, organisations that store personal data on their computer systems have a legal obligation to keep that data secure. If personal data is stolen during a cyberattack and the company can be shown to have been negligent in terms of keeping the personal data secure, then the company could face prosecution under the Data Protection Act and receive a large fine.

Forensic research requirements to identify data lost, stolen or copied

Research

Computer forensics is a complex and fascinating topic. Research the tools and techniques which are use.

If an organisation detects that a serious IT security breach has occurred, such as its network being hacked and data stolen, then the time, effort and expense of investigating the incident can be considerable and may seriously disrupt the day-to-day running of the organisation. In order to preserve forensic evidence of the attack, it is likely that system would need to be shut down and the hard discs may need to be removed for detailed forensic investigation. This would require the installation of new discs and the restoration of the organisation's data from backup copies. The analysis of the discs and other evidence of a cyberattack is a complex and highly skilled process, and it is likely that the organisation would need to contract in cybersecurity experts to do the work, which would, of course, add to the cost of the security breach.

PAUSE POINT Carry out research into recent IT security issues at large companies. What actually happened? Which type of cyberattack described in this section was used? What was the impact on the company? Did they lose money or face legal consequences?

Hint News websites, such as the BBC news website or newspaper websites such as the Guardian or Telegraph Technology sections (www. telegraph.co.uk/technology or www.theguardian.com/uk/technology), are excellent places to start your research.

Extend Consider how the company could have avoided the security breach. Were there protection methods they could have used? If there were, why did they not use them? How can organisations protect themselves in the future from this kind of security issue?

B Investigate cryptographic techniques and processes used to protect data

Cryptography is the science of hiding data in such a way that, although others may be able to see the data, only the intended recipient can understand it. Cryptography pre-dates the computing era and examples of its use to protect secrets can be found going back thousands of years. Cryptanalysis was the main use of computers when they were first created, and the need to crack enemy **cyphers** during World War II spurred the development of computers. Colossus, the world's first fully programmable digital computer, was built at Bletchley Park in the UK during World War II to crack coded German messages.

Key term

Cypher – algorithm used to encrypt and decrypt data (alternative spelling is cipher).

Cryptographic principles

In this section, you will look at the principles and uses of encryption. There are two main principles of modern computer-based encryption: integrity and confidentiality.

Integrity

The principle of **integrity** provides confidence that data has not been modified in any way. Integrity of data is achieved by creating a value called a 'hash' (sometimes called

a 'checksum'). The hash is sent with the data. When the data is received, the hash is recalculated and the result is compared with the hash that was sent with the original file. If both of the hashes are the same, then you know that the data has not been tampered with.

Key term

Integrity – principle of being honest and having strong moral standards.

Link

Hash functions are described in more detail in the section 'Cryptography methods'.

Confidentiality

Data kept **confidentially** can only be viewed by authorised users, that is, it is kept secret from anyone who is not meant to know it. Encryption is used to scramble data, making it unreadable without a key.

Key term

Confidentiality – principle of keeping something a secret. If you are told some information in confidence, then you must keep it a secret.

Uses of encryption

There are various of types of encryption that are used for a wide variety of different applications, some of which are described in the following sections.

Digital rights management

The ease with which digital media can be copied has given the music, video and games industries a problem. They need to find a way to allow legitimate purchasers of copyright material to have access to it while preventing it from being copied and sold on illegally. The way in which they try to do this is through digital rights management (DRM). There are many different DRM systems in use, not all of which use encryption. An encryption-based DRM system will typically encrypt the media and then the decryption key is provided only to authorised users of the media. An example of this kind of DRM is the CSS (content scrambling system) used on DVD video discs. This system is designed to ensure that only licensed DVD players can play that commercial DVD video.

Password storing and salts

Passwords are widely used to authenticate users on a wide variety of systems. In order to authenticate a user, the user's passwords must be stored on the system, but storing them as plain text would be an obvious security risk, so passwords are therefore encrypted. However, if an attacker has access to the password file and knows the cryptographic function that is used to produce the encrypted passwords, then they could make a list of many commonly used words and encrypt them using the same function and see if that matches any of the encrypted passwords in the password file. The attacker will then know at least some of the passwords because they have matched them. (There are a lot of encrypted password files that use lists of common words available on the internet, which attackers can get access to). This kind of cyberattack is sometimes called a dictionary attack.

'Salts' are added to the encrypted passwords to make them more difficult to crack. A salt is simply a random value added to an encrypted password so that it will not match the encrypted passwords that an attacker generates with their dictionary attack.

Obfuscation and steganography

Obfuscation and **steganography** are not strictly cryptographic methods but they do rely on hiding secret data in amongst other non-secret data (in a 'carrier' file). For example, files containing secret information can be hidden inside image files. Image files are well suited for steganography because they are large and so hiding a relatively small amount of text inside them by altering a few of the pixels within the image would not be noticeable.

The benefit of steganography compared with cryptographic methods is that this method does not attract attention because the image appears to be innocuous. The main application of steganography is the requirement to send information over the internet to individuals without raising the suspicions of others. Programs use steganography to hide secret data in carrier files that are freely available on the internet.

Key terms

Obfuscation – means to make something more difficult to understand, that is, to make it obscure, unclear or unintelligible.

Steganography – is the practice of concealing information within other non-secret data.

Secure transactions

The basis of e-commerce is the ability to send sensitive financial information securely over the internet. When

making a credit card purchase from an e-commerce website, it is important that the transaction details are secure, otherwise a third party may be able to intercept them. Secure transactions are achieved by encrypting the data sent from the user's internet browser to the web server that hosts the e-commerce site.

Link

How secure transactions are achieved is explained in more detail in the section 'Public-key encryption', in 'Cryptography methods'.

Two-factor authentication

In a typical home computer system, authentication just involves a single factor: the user password. Two-factor authentication involves both something that the user knows (such as a password or a PIN number) and also something that the user has in their possession (such as a USB memory stick which contains a security token or a smart card). Security tokens come in a variety of types including those that display a value that the user must enter into the computer system to authenticate themselves and those that are plugged into a USB port. Some banks provide customers with card readers that are used in conjunction with their debit card to provide a security code that can be used to access online banking facilities.

File, folder and disc encryption

Data held on a computer is protected by the operating system, which only allows authorised users to access it. However, if the hard disc is removed from the computer and connected to a different computer, then the operating system protection can be circumvented and the data can be accessed by others. This is a particular issue with laptop computers which can more easily be stolen or lost. One way to help protect data is to encrypt it so that it is only accessible to whoever has the encryption key. Microsoft® Windows® operating system has an encryption system for individual files and folders called the encrypted file system (EFS). Files or whole folders can be encrypted with a key associated with the user account of the user who encrypted them. For that user, the files are transparently encrypted and decrypted. For anyone else, the files are inaccessible.

Microsoft® also provides a system called BitLocker with some versions of Windows® operating system. This is a full disc encryption system that encrypts the whole hard disc (in practice, this means that it encrypts the whole laptop). The system works in conjunction with the trusted platform module (TPM) hardware included in many motherboards, which stores a cryptographic key. This key is used to encrypt the hard disc and can also be combined with a password which the user must enter when the device boots up. If the hard disc is removed from that computer and connected to another, the TPM key will be different and so the files cannot be decrypted.

Encryption of communication data

Mobile telephone conversations are transmitted using encrypted digital data. GSM (global system for mobile communication) mobile phone conversations are encrypted using the A5/1 stream cypher. The A5/1 stream cypher is no longer secure and it has been shown that it is possible to crack the encryption and decrypt mobile phone data, allowing conversations to be eavesdropped on in real time. Information leaked by Edward Snowden showed that the USE National Security Agency (NSA) can decrypt A5/1. Mobile phone voice messages have also been widely 'hacked' by unscrupulous journalists, although these are not encrypted and social engineering methods were used to access this data.

Research

The use of the A5/1 cypher to encrypt mobile phone communications is interesting to look into. Why, for example, was a relatively short key length of 54 bits used rather than a more secure longer key? Are there other methods that can be used to protect mobile phone communications?

Legal and ethical issues

Encryption can be used for a range of purposes, some of which raise legal and ethical issues. For example, those intent on breaking the law, such as criminals and terrorists, can use encryption to hide their communications from law enforcement agencies. Before computer data encryption became widely used, encryption technologies were closely guarded secrets and were not available for the general public to use. The widespread availability of encryption techniques to the general public means that anyone can now hide data from prying eyes, both for legitimate and for unlawful/unethical purposes. Some argue that individuals should be allowed to keep their data secure, while others argue that technology companies should build a 'backdoor' into their products to allow law enforcement and other government agencies to access it if needed. The risk with doing this is that criminals and terrorists may then also be able to access these 'backdoors'.

Case study

The San Bernardino shooting

In December 2015, 14 people were killed in a terrorist attack in San Bernardino, California, US. An Apple iPhone® 5C belonging to one of the terrorists was recovered, and the FBI wanted to unlock the phone to see if other people had been involved in the attack. (The two terrorists who carried out the attack were shot dead by police.)

However, back in 2014, it had been revealed that the FBI and the British GCHQ had ways of accessing all information on Apple® and other smartphones. In response to this, Apple® had improved its encryption in iOS® version 8, and this prevented the FBI gaining access to the terrorist's phone. The FBI therefore asked Apple® to unlock the phone but Apple® refused, stating that it was company policy not to undermine the security features of its products as to do so would not be in the interests of its customers. The FBI therefore issued a court order compelling Apple® to unlock the phone. However, before the case came to court, the FBI dropped the case because they stated that a third party (said to be the Israeli company Cellebrite) had enabled them to access all the data on the phone.

The case raised many technical and ethical questions about whether technology companies should build a 'backdoor' into their encryption products for the purpose of allowing government agencies to access them in such cases, or whether it is in their customers' interests to protect their encryption methods at all costs.

Think about it

1. Do you think it was right for the FBI to want access to the information on the terrorist's phone?
2. Why would Apple® want to protect its encryption methods and do you think they were right to withhold the information?
3. Should the government have access to our data? After all, if you are doing nothing illegal, what is there to hide?

Computational hardness assumption

Most modern cryptographic methods are not impossible to crack. Instead, they rely on being very hard to crack, given the computing power available. In other words, they rely on the fact that it would take a computer a very long time indeed to crack the encryption and therefore no one would bother trying to crack it. However, the power of computers continues to increase. In theory, a point may be reached in the future when current encryption methods can be overcome in a relatively short time. At such a time, current encryption methods would stop being useful because it would become very easy to crack them.

Cryptographic methods

There are a number of different methods and standards that can be used to encrypt data, which reflect the various different applications for which encryption is used. This section looks at the most important cryptography methods.

Shift cyphers

The use of a shift cypher is one of the oldest and simplest methods of encryption, pre-dating computers. It simply involves substituting one letter of the alphabet for another. Table 7.1 shows an example of a shift cypher where the letters are shifted two to the right in the alphabet.

Letter of the alphabet	A	B	C	D	E	F	G	H	I	J	K	L
Substituted letter	C	D	E	F	G	H	I	J	K	L	M	N

Table 7.1: Shift cypher – only the first 12 letters are shown, but you can easily write out the complete table yourself to follow the example and encrypt your own messages.

So in the example in Figure 7.1, the words

IT SECURITY

become KV UGEWTKVA when encoded.

While shift cyphers, as a method of cryptography, are easy to understand, they are also very easy to crack and provide almost no security.

One-time pad

While a shift cypher offers very little security, a one-time pad cannot be cracked if used correctly. Use of a one-time pad involves each character of the text to be encrypted being combined with a different random number. The list of random numbers used to encrypt the message is called a 'pad', and each pad is only used once. The pad must be as long as the message to be encrypted. The one-time pad method works as follows.

1. The two people who wish to transfer a message must both have a copy of the same pad.
2. The sender combines each character of the message to be sent with the corresponding character on the pad (that is, the first character of the message is combined with the first character, the second character with the second character on the pad and so on).

3 Once the message is encrypted, the sender destroys their copy of that pad.
4 The encrypted message is sent to the recipient.
5 The recipient decodes the message using their copy of the one-time pad and then they also destroy the pad.

Table 7.2 shows how the character in the one-time pad is used as the shift that is applied to the corresponding letter of the message. To determine the shift to be applied, A = 1, B= 2, C = 3 and so on. This is similar to the shift cypher but every letter is shifted by a different amount.

Table 7.2: One-time pad example

One-time pad	E	Y	B	R	G	K	M	F	E	L
The shift	5	25	2	18	7	11	13	6	5	12
Message	I	T	S	E	C	U	R	I	T	Y
Encrypted message	N	S	U	W	J	F	E	O	Y	K

Link

An internet search for 'one-time pad' will provide a lot more information on this topic.

One-time pads are not used widely because of the problem of transmitting the pad to the intended recipient. For example, if you sent the encrypted message over the internet to the recipient and then sent the pad, anyone who intercepted both the message and the pad could decrypt the message.

So the pad must be passed to the recipient securely, not using the same method as the message, which makes it impractical if you want the message to stay secret. If you have the pad, then the message is easy to decrypt – without the pad it is impossible to decrypt.

Hash functions

A cryptographic hash function takes an input of variable length and returns a fixed-length value. The value produced by a hash function is often called the 'digest' or 'hash value', and it acts as a kind of 'signature' or 'fingerprint' for the input file. The main purpose of hash functions is to ensure that a message has not been altered in any way during transmission (i.e. that its integrity has been maintained).

A good hash function has the following characteristics:

- it is relatively easy and fast to compute
- it is very difficult to reverse (to work out the message given the hash value)
- it is very unlikely that two messages will generate the same hash value.

There is a range of different hash functions that have been widely used, two of which are detailed here.

- **Message digest (MD):** There are various versions of this hash function. MD5, for example, was often used to check the integrity of downloaded files. However, in 2004, it was found that MD5 can be cracked quite easily, so it is no longer recommended for use.
- **Secure hash function (SHA):** Again, there are a variety of versions. SHA-1 was widely used in a number of applications including secure socket layer (SSL) communications, which are the basis of all secure internet transactions. Weaknesses were discovered in SHA-1 in 2005. SHA-2 and SHA-3 are the current versions in use by SSL and other applications.

Worked Example: Create your own hash

It is quite easy to create a hash for any file using free software that is available on the internet. Google 'create a hash or create a SHA1 hash' to find websites where you can do this.

You could use a product like this if, for example, you ran a small software house that provides updates and patches for products for your customers to download and install. By creating hashes for each of your downloadable updates and sending them to your customers, you can provide a means for them to check that the downloads have not be compromised in any way and only contain the data you intended (i.e. that their integrity has been maintained).

Block and stream cyphers

Block and stream cyphers are different techniques for encrypting data.

A block cypher will take the data that is to be encrypted and break it up into blocks of a fixed number of bits (for example, 64 or 128 bits are commonly used), while a stream cypher will encrypt each digit (e.g. a bit) individually. Stream cyphers are better suited to applications where the size of the data is unknown or it is sent in a continuous stream such as when streaming audio or video. Block cyphers are better suited to situations in which the size of the data is known, such as when encrypting a database file.

Encryption algorithms

The basic idea behind encrypting data is that the data to be encrypted is taken with a key and passed through a process that makes the original data unreadable. At the receiving end, the key can be used to decrypt the data making it readable again. Where the same key is

used to both encrypt and decrypt the data, this is known as **symmetric key encryption**. **Asymmetric key encryption** uses two matched pairs of keys.

Key term

Symmetric and asymmetric key encryption – in symmetric key encryption, the same key is used to both encrypt and decrypt the data, whereas, in asymmetric encryption, two keys are used, one to encrypt the data and another, different but mathematically paired, key is used to decrypt it.

There are many different encryption algorithms, also known as encryption standards. Some of the best known are detailed here.

DES

DES (data encryption standard) was developed in the 1970s by IBM and was once the primary symmetric key encryption technology. It is a block cypher system with a key length of 56 bits. DES is now considered insecure as its short key length allows it to be cracked quite easily. This example highlights how, in the 1970s, there was insufficient computing power available to crack DES in a reasonable amount of time, but, as computing power has increased, it has become possible to crack the encryption. DES is believed to have been cracked for the first time in 1999, in 22 hours 15 mins.

3DES

3DES (triple DES) was developed using the same algorithm as DES but using a 'key bundle' of three keys for each of the 56 bits. Currently, 3DES is secure and has not yet been cracked. 3DES is used to secure data used to make financial transactions by credit/debit card.

RSA

RSA is an asymmetric public-key system that is widely used for secure internet transactions. The system is named after its inventors Ron Rivest, Adi Sahmir and Leonard Adleman of the Massachusetts Institute of Technology (MIT) in the US.

Link

The RSA system is described in detail in the section 'Keys and digital certificates'.

AES

AES (advanced encryption standard) is a strong block cypher that uses 128 bit blocks and is often used in symmetric key encryption as it is very secure. It was adopted by the National Institute of Standards and Technology (NIST) in the US after a long evaluation of various different encryption algorithms. AES can use encryption keys of 128, 192 or 256 bits, and is sometimes called 'AES-128', 'AES-192' or 'AES-256' depending on the key size used. The more bits in the key, the harder it is to crack it. AES is popular, not just because it is strong, but also because it is efficient and requires less computational power to encrypt/decrypt files than many other algorithms.

Mathematical principles

The mathematics behind encryption is very complex. However, there are some basic principles that you can look at.

Cryptographic primitives

Most modern computer-based cryptography is based on a set of algorithms or building blocks which are sometimes called cryptographic primitives.

One-way functions and integer factorisation

The idea behind a one-way function is that it is easy to compute but very hard to reverse. Hash functions are an example of one-way functions. A simple example of a one-way function would be to take two **prime numbers** and multiply them together. This is easily done but, if the number is very large, there is no quick way of decomposing the resulting number back into the original values (there is no easy way of predicting which two prime numbers were multiplied together). This problem is known as integer factorisation and, as yet, no algorithm exists to factorise large (hundreds of bits) numbers quickly.

Key term

Prime number – a number that can be divided by only itself and 1. There are lots of prime numbers such as 3, 5, 7 and 151.

Worked Example: One-way function

Here is a simple example of a one-way function.

1. Take the number 147. Which prime numbers can be multiplied together to make 147?
2. One way to work this out would be to work up through the prime numbers starting at the smallest, which is 2.
3. 2 is the smallest prime number but 147 is not an even number so 2 will not divide into it.
4. The next prime number is 3.
5. 147 ÷ 3 = 49

6 49 cannot be divided by 2, 3 or 5 but it can be divided by 7
7 49 ÷ 7 = 7
8 As 7 is a prime number, you do not need to go any further.
9 So you now have all the prime factors of 147, they are: 3 and 7.

While this might seem quite simple for a small number, for numbers with hundreds of bits it would take months of computational time to find all the factors. As no one wants to spend months working out the factors, using a one-way function for integer factorisation is quite secure.

Link

For more about integer factorisation of prime numbers, see http://www.mathsisfun.com/prime-factorization.html

Pseudorandom functions

Pseudorandom functions are types of programming function that create apparently **random numbers** efficiently. These are required by many cryptographic methods including generating keys to encrypt data and creating cryptographic salts to be added to passwords. To be regarded as a secure pseudorandom number generator, the function needs to produce a stream of bits for which there is no known method of predicting the next bit to be produced with a probability of success that is greater than 50 per cent.

Link

For more about cryptographic salts and their use in storing passwords, look back at the section 'Password storing and salts', in 'Cryptography methods'.

Key term

Random number – a number in a list of numbers that is not related to the previous or to the next number in the list.

Applications of cryptography

There are several types and applications of cryptography, many of which are covered in this section.

Keys and digital certificates

Symmetric key encryption is useful in some software applications but is not an effective security solution when you want to securely send data to another person, as they will need the encryption key to decrypt the data. Sending the key to the person would create an obvious security risk because, if the encrypted data and the key were intercepted by a third party, then they would be able to decrypt the message easily.

To get around this problem, a more complex system has been developed called public-key encryption, which is based around the RSA system.

Public-key encryption works like this.

1 A pair of mathematically related keys are generated – one is the public key and the other is a private key.
2 The public key is available to anyone, but the private key is kept secret.
3 The public key is used to encrypt messages.
4 A message encrypted with the public key can only be decrypted using the private key.

Public-key encryption is complex, so trying to understand it is best done with an example. SSL (secure sockets layer) or the more up to date TLS (transport layer security) are widely used implementations of public-key encryption. Public-key encryption using SSL or TLS is part of the HTTPS protocol, which is used when you access a secure internet website such as a banking site or when you are paying for a purchase on an e-commerce site. SSL and TLS use a **digital certificate** provided by a **certificate authority** to prove that the website you are communicating with is a trusted site.

Key terms

Digital certificate – to run a secure website, the site owner must apply to a certificate authority to obtain a digital certificate. The digital certificate certifies that the website is genuine and owns a particular public key. Without a digital certificate, it might be possible for a bogus website to impersonate a secure site.

Certificate authority – a certificate authority (CA) is an organisation that issues digital certificates. There are many commercial CAs including Comodo®, Symantec and GoDaddy®. There is also a free CA called Let's Encrypt™ which is sponsored by organisations such as Cisco® and Google Chrome™ browser.

The process that takes place when your web browser logs on to a secure site which has a digital certificate is outlined in Figure 7.1.

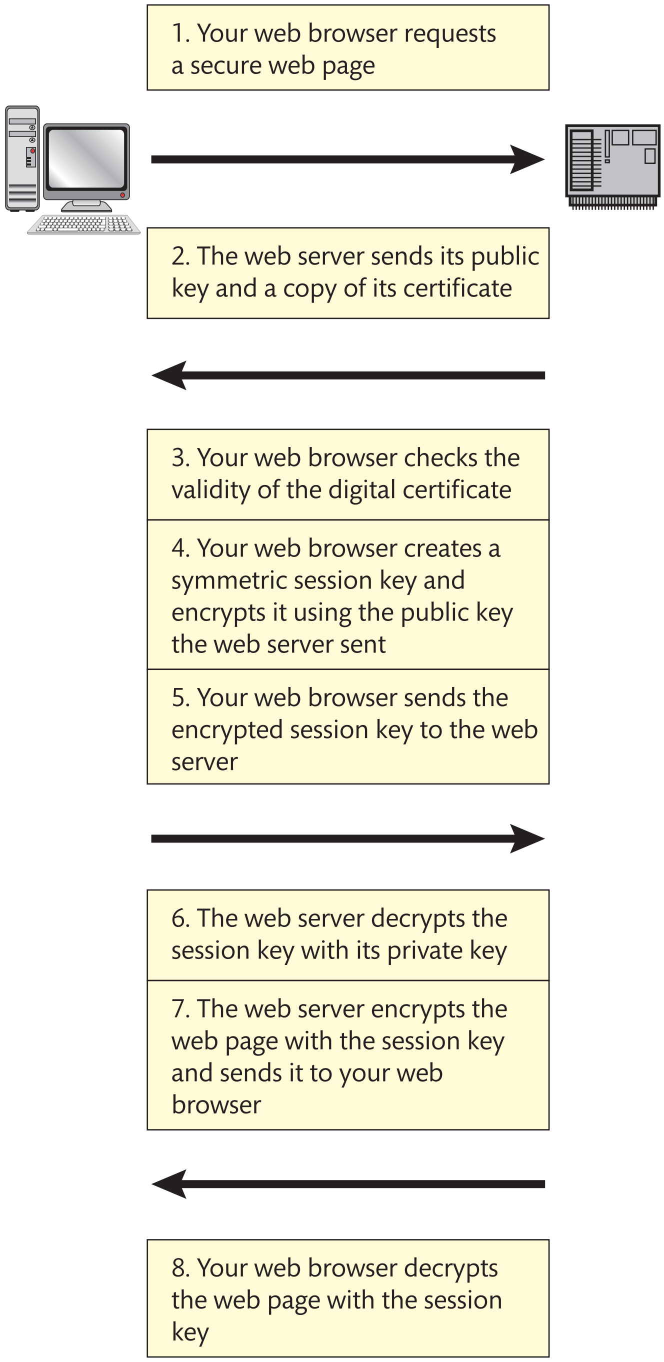


Figure 7.1: Public-key encryption process

As it is not efficient to encrypt large amounts of data using asymmetric key encryption, it is only used to transfer the symmetric session key between the client and the server. This gets over the problem of sending a symmetric key. The rest of the session data is encrypted using the same symmetric key.

Using this method, it is possible to exchange a symmetric key without it being possible for a third party to intercept it. This is known as a 'Diffie–Hellman key exchange', named after the two American cryptologists Martin Hellman and Whitfield Diffie who developed the technique.

Worked Example: Setting up a secure site

Suppose you wanted to set up an online shop and needed to provide secure facilities to enable your customers to pay safely with credit/debit cards. How would you go about doing this?

1 First you would need to create a public and private key pair. There are many programs available that allow you to do this, for example PuTTYgen.

2 Then you would need to create a certificate signing request (CSR) which provides information about the website, you and your public key. Popular web server software such as APACHE® and Microsoft IIS® provide tools to create CSRs.

3 The CSR is sent to a certificate authority.

4 The certificate authority verifies your website's details and then creates a certificate which includes your public key.

5 Your digital certificate is then installed by your web server and, whenever anyone requests a HTTPS session with your web server, it sends the certificate and the public key.

You must keep the private key safe. If the private key is compromised (if an attacker obtains a copy of it) the system is no longer secure and the associated certificate must be revoked and new keys and certificates created.

Certificate authorities such as Verisign, GeoTrust and Thawte have their trusted root certificates included in popular web browsers. This means that any certificates issued by these certificate authorities will be trusted by your browser. You can see the root certificates included with your web browser. The trusted root digital certificates for Internet Explorer are shown in Figure 7.2.

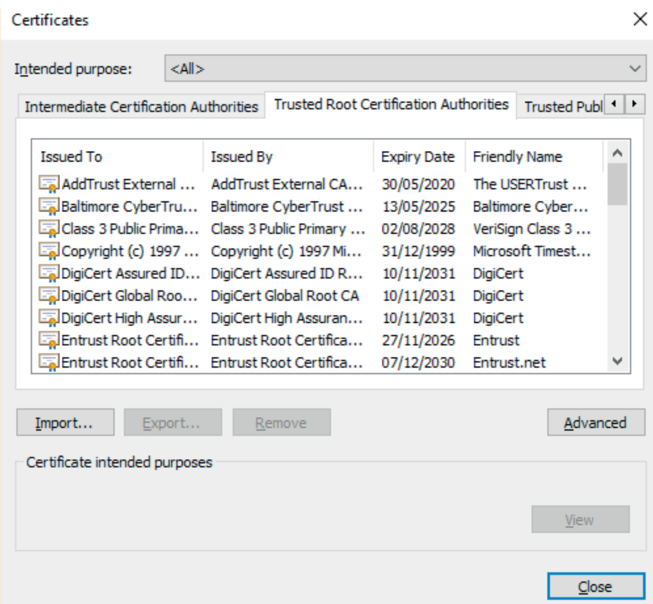


Figure 7.2: Trusted root digital certificates for Internet Explorer

Virtual private networks (VPNs) and generic routing encapsulation (GRE) tunnels

A VPN allows a private network (such as an organisation's LAN) to be extended across a publicly shared network, usually the internet. It allows users to access the organisation's LAN (and therefore applications and shared data) when they are remotely connected. (Essentially, it allows an employee to access their company computer network at home as if they were at the office.)

Naturally, connecting users by a VPN creates security issues since private data is being transmitted over a public network. VPNs usually include an aspect of user authentication to ensure that only authorised users can access the VPN and data is encrypted so that all of its interactions are unreadable. VPNs use what is called a 'tunnelling protocol' which creates a virtual tunnel in which the data travels across the internet. The internet protocol called IPsec is often used for VPNs as it encapsulates and encrypts data packets and also provides facilities for hosts to authenticate each other before starting data transmissions.

Generic routing encapsulation (GRE) is a tunnelling protocol developed by Cisco® Systems for, amongst other things, the implementation of VPNs. GRE works by wrapping an inner data packet in an outer IP packet (rather like taking a letter, placing it in an addressed envelope and then putting that envelope in another addressed envelope). This allows the creation of a VPN over a wide area network (WAN) such as the internet. The routers on the WAN ignore the inner packet and process the packet based on the information in the outer packet. Only when the GRE packet reaches the relative safety of a private LAN is the outer packet stripped off.

Encryption of data on WiFi networks

WiFi networks use radio waves to transmit data, which, because they are broadcast media, can be picked up by anyone in range of the radio signals. It is therefore essential that the data sent over WiFi networks is encrypted. A number of different encryption methods can be used with WiFi networks.

WEP (wired equivalent privacy)

WEP (wired equivalent privacy) was the original encryption method used for WiFi networks. It used a stream cypher with a 64 or 128 bit key, usually entered as 10 or 26 hexadecimal digits. In 2001, WEP was found to be easily cracked and so is no long regarded as secure.

WPA (WiFi protected access)

WPA was developed in response to the weaknesses in WEP and one benefit of it was that it did not require new hardware (routers which had been designed for WEP could use WPA with just a software upgrade). An improved version of WPA called WPA2 was launched in 2006. WPA uses a more advanced form of encryption than WEP called TKIP (temporal key integrity protocol), but it was developed as an interim replacement for WEP after its vulnerabilities had been discovered, and it was not intended as a permanent solution.

WPA2 was developed as a permanent replacement for WEP. WPA2 uses a much stronger form of encryption than WPA called CCMP (cypher block chain messaging authentication code protocol) which is based on the AES encryption standard. In order to keep WPA2 as secure as possible, it is recommended that a key of at least 20 characters, with a mix of upper and lower case characters and numbers, is used.

Personal versus enterprise modes

Most home-based WiFi routers use WPA2 personal mode where access is provided through a **pre-shared key (PSK)**, which is entered into any device that a user wants to access the network. This does not provide any authentication of the users as anyone who has the key can access the network. In most cases, the keys are printed on a label attached to the router.

Alternatively, enterprise mode can be used, which requires authentication of users and so is more secure. To set up enterprise mode, WPA2 requires that WiFi users provide unique credentials to access the network. To implement this, a server is used which holds a database of valid usernames and passwords. To gain access to a WPA2 enterprise WiFi network, a user must enter their valid credentials to be authenticated by the server.

Key term

Pre-shared key (PSK) – a cryptographic term which describes an encryption key that is provided to authorised users to allow them to encrypt and decrypt data.

Research

Bluetooth is widely used for the short range transmission of data between mobile devices. Although Bluetooth data packets are encrypted, there are a number of security issues when using Bluetooth. Find out how the data packets are encrypted and research the security issues involved when using Bluetooth.

Reflect

Assessment practice 7.1 includes a task in which you need to evaluate the security techniques that could be used. In an evaluation, you need to consider both the positive and negative aspects of the topic under consideration and produce high quality, justified recommendations and decisions and reach a considered conclusion. Your evaluations are essential to achieving the highest grade for this unit and to maximise the quality of your work.

Assessment practice 7.1

A.P1 A.P2 A.P3 B.P4 A.M1 B.M2 AB.D1

You have recently started working for a medium-sized company in the IT support department. The directors of the company have become increasingly concerned about IT security threats, so you have been asked to give them a presentation on the topic. They are also interested in how encryption can be used to protect the company data. Your presentation needs to cover the following.

- An explanation of what the different IT security threats are.
- An explanation of the principles of information security and the legal issues related to IT security.
- An explanation of the principles of cryptography and an analysis of how it can be used to protect data and the impact it can have on security.
- An analysis of the impact that IT security threats can have on the company's business, taking into account information security and legal issues.
- An evaluation of the techniques that could be used to protect the company from IT security threats while taking into account the issues of information security and legal requirements.

Plan

- What is the task? What am I being asked to do?
- How confident do I feel in my own abilities to complete this task? Are there any areas I think I may struggle with?

Do

- I know what I am doing and what I want to achieve.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.

Review

- I can explain what the task was and how I approached the task.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

C Examine the techniques used to protect an IT system from security threats

As there are so many different security threats to IT systems there needs to be a large range of precautions put in place to defend these systems. This section will examine the techniques used to protect IT systems from security threats.

Physical security

This section will look at the physical security of computing systems, including building security, backing up data and IT disaster recovery plans.

Building and computer/network room security

Computer systems, especially servers, need to be protected from physical access because theft of such valuable equipment, especially hard drives, is a serious security risk.

Tip

Laptop computers and other digital devices can be secured to desks by cable locks to help deter casual theft.

Servers, routers and switches (network equipment) should be kept in secure locations with carefully controlled access. Ideally, access should controlled by proximity key-card entry, whereby the door lock will only open when someone with the right kind of key-card approaches the door. These types of electric door lock can be connected to a system that maintains records of who entered the computer room and when. This data can prove useful if an investigation is needed into when a security breach took place. (Although these electric door locks can tell you which key-card was used, they cannot unfortunately ensure that the person using the key-card was actually the person authorised to have that key-card. The key-card could be stolen.) Thought should be given to the location of a server room within a building. Ideally it should not be on the ground floor or have windows or doors which are accessible from outside the building.

In a large building, as well as the main server room it is common for each floor to have a network room where routers, switches and other network equipment is located. These rooms should be kept locked and only authorised personnel should have access (they could use key-card entry). This level of security can help to prevent the installation of illicit network monitoring equipment.

If a very high level of security is needed, **biometrics** can be used. As mentioned above, one problem with traditional key-cards is that they can be lost or stolen themselves. Biometric authentication gets around this problem because it uses data about the person themselves such as their voice pattern or fingerprints.

Key term

Biometrics – the use of biological data for identification, for example voice control, facial recognition, fingerprints or iris scans. Biometrics uses features of a person that are unique to the individual. These bits of personal information are virtually impossible to forge. Biometrics also have an advantage over other identification methods because the individual does not have to keep anything (such as an ID card) or remember anything (such as a password). DNA identification technology is in its infancy but the idea of using a person's DNA to uniquely identify them has potential for future use in the IT security industry.

One problem that can occur with all of these door lock access systems is tailgating. This is when one person (who is not authorised to enter an area) closely follows another person who is authorised. Out of politeness, the first employee will almost always allow the unauthorised tailgater to follow them through the door. Tailgating can be prevented by installing turnstiles which only allow one person to pass at a time.

Many organisations are making increased use of electric surveillance. Closed circuit television (CCTV) is commonly used to monitor the outside of a building and also, in some cases, high-risk areas inside a building, such as the entrance to a server room. CCTV provides a helpful visual deterrent as the images can be recorded and provide useful evidence for investigations which could lead to prosecutions.

Discussion

CCTV is widely used in public places, and you may well have CCTV installed in parts of your school or college. With a peer or in a small group, discuss these questions. Do you think that this is an invasion of your privacy? Is there too much CCTV? How are the images recorded by CCTV systems used? Is there a possibility that these images could be abused?

Backing up data

As it is not possible to guarantee that the security of a system can always be maintained, it is vital that regular **backups** are made. In a situation where a malware infection has deleted or damaged data in an IT system, it may be necessary to recover the system by using data that was backed up prior to the infection. Since some time may pass between the security breach (when the system is compromised) and its discovery, it is necessary to keep previous backups for several weeks. When deciding upon a backup regime for an organisation's computer systems there a number of factors that need to be considered.

Key term

Backup – a copy of data files that is kept in case the original files are lost or damaged.

How often should the backup be performed?

Deciding how often a backup should be performed can be a difficult choice for an organisation. It is common to use the concept of a recovery point objective (RPO) to define the maximum amount of time a business can afford to lose data from an IT system. Many organisations with non-critical systems use a daily backup, but this can only provide RPO in the region of 24 hours. If an organisation requires an RPO shorter than this, then they may need to consider using online backup methods by which data is constantly backed up to a remote website via a network connection.

What media should be used for the backup?

Traditionally, magnetic tape has been used for backup and it still provides a low-cost solution which, due to the sequential nature of tape, can write data to the tape very

quickly. Alternatives include backup to hard disc or optical media such as DVD-ROMs. The choice may be related not just to cost but also to the amount of data that needs to be backed up. Optical media, in particular, are limited to relatively small sizes as the maximum capacity of, for example, a Blu-ray disc is 128GB.

Selection of data for backup and storage of backup media

There are a number of different options for selecting the data to be backed up. In many cases, it is not practical (due to the large amount of data involved) or necessary to complete a full system or system image backup regularly. It may not be necessary to back up all the data because much of it may be static, that is, it does not change very often.

An incremental backup is a method whereby a full backup of all the required data is done first (for example over the weekend) and then a daily backup is done each day of the week which backs up only those files that have changed that day. This means that each backup contains less data and can be done more quickly. The disadvantage of this system is that, assuming you do a full backup only every weekend and incremental backups each weekday, if there is a hard disc failure on a Thursday, then, to restore all the data, you need to restore the weekend's full backup and then all the incremental backups from Monday through to Wednesday.

An alternative to an incremental backup is a differential backup. With this method a full backup is done at the start and then each subsequent backup backs up any changes since the full backup. This could again be implemented with a full backup at the weekend and then daily differential backups. Each daily backup would require progressively more time and storage since it backs up all changes since the weekend's full backup, not just since the previous day (as with the incremental backup). The benefit of this method comes if you need to restore the backups. With differential backups, if a failure occurred on a Thursday all you would need to do is restore the weekend's full backup and Wednesday's differential backup.

Tip

Whichever method of selecting data for backup is used, the backup media must be stored offsite to provide protection against disasters such as fire or flood.

Tables 7.3 and 7.4 show how incremental and differential backups work.

Table 7.3: Incremental backup

Weekend	Monday	Tuesday	Wednesday	Thursday	Friday
Backup everything					
	Backup changes since yesterday				
		Backup changes since yesterday			
			Backup changes since yesterday		
				Backup changes since yesterday	
					Backup changes since yesterday

Table 7.4: Differential backup

Weekend	Monday	Tuesday	Wednesday	Thursday	Friday
Backup everything					
	Backup changes since full backup				
		Backup changes since full backup			
			Backup changes since full backup		
				Backup changes since full backup	
					Backup changes since full backup

Tip

Never keep your school/college assignment work in just one place. Make backups at regular intervals (at least weekly – more often if you have done a lot of work on them).

IT disaster recovery plans

Many organisations rely on IT systems to run their businesses, so they need to consider how they would continue to operate if the systems were destroyed or rendered unusable by fire, flood, terrorist action or other disaster. In these situations, data backups alone are not sufficient because, if there is no IT system to run the applications on, the data backups are of little use. Planning to deal with such eventualities must be done prior to the event happening and there are a number of approaches that can be taken to disaster recovery.

When looking at backups, you were introduced to the concept of RPO, which is related to the amount of data that a business can afford to lose. When considering disaster recovery, the concept of recovery time objective (RTO) is also important. RTO is the target time in which a business wants to recover its systems following a disaster such as fire or flood. For example, an organisation that sells its products through a website might generate £10,000 in revenue every hour, so a 12 hour outage would cost the company £120,000. Therefore investing significant sums of money in preparing for a disaster would be considered money well spent. Without pre-planning, the recreation of the IT systems following a disaster could take weeks or even months. The shorter the desired RTO, the more investment an organisation needs to make to prepare for a potential disaster. There are a number of approaches that can be taken to disaster recovery.

Hot site

A hot site is where the company maintains a complete working duplicate of all its server systems at a geographically separate site to its main computer operations. This includes all computer and networking equipment and the internet connections required to run the systems. In large organisations which require very short RTO (banks, for example), the hot site will often be a company location where some non-critical computing functions are carried out but which has the capability to take over from the main site at very short notice. Obviously, a hot site is expensive to set up and maintain as it more or less doubles the cost of running the IT systems.

Cold site

A cold site is essentially just a building with suitable power and connectivity. In the event of a disaster at the main site, the company would need to purchase servers and other equipment and set the systems up using backups for system configuration and application data. A cold site has a much longer RTO than a hot site – probably weeks unless the systems are very straightforward – but it is relatively cheap and the organisation can rent the cold site from a company that provides this service. Cold sites can be shared by many different organisations.

Warm site

Often a hot site is too expensive for many organisations to run and a cold site takes too long to set up. A warm site provides a compromise, by having hardware in place but not fully configured and running. As with a cold site, there are companies that specialise in providing ready-to-use warm sites. Organisations that use these services can do a 'test run' by bringing their backup tapes and trying out a set-up of the system so that they can have all the issues ironed out before disaster strikes. Of course, since configurations and software are likely to change, the company would probably need to do occasional test runs at the warm site to check that everything still works.

Policies and procedures.

Organisational policies and procedures do not provide any direct protection from security threats. Instead, they make employees aware of the risks and define the sorts of unsafe practices that they should avoid.

Organisational policies and their application

Organisational policies on acceptable IT use should include the following:

- Internet usage guidelines: These guidelines should cover ways to avoid inappropriate websites that might harbour malware and to avoid security breaches when downloading files.
- Email usage policies: These should give guidance on treating emails from unknown sources with care, including not opening attachments or following web links.
- Security and password procedures: These procedures should outline how to keep passwords secure, for example by not writing them down or sharing them with others.
- Staff responsibilities: These responsibilities might include things like locking desk drawers and filing cabinets, locking or logging off unattended computers, wearing ID badges, challenging strangers, not giving out information over the phone and other general security procedures designed to protect against social

engineering threats. It is likely that organisations will also have a set of rules relating to BYOD, in terms of what staff are and are not allowed to do with their own devices on work premises and how they are allowed to connect their devices to the organisation's systems.

- Disciplinary procedures: Staff should be made aware that deliberately breaking rules in any of the organisation's IT usage policies will have disciplinary consequences, which should be described in the staff disciplinary procedures document.

Just having a policy written down is not by itself any use, so staff must be made aware of the policies and be trained in safe IT procedures. As new IT threats appear all the time, staff should be regularly updated to raise their awareness of new threats and how to avoid them.

Security audits

Having a set of IT usage policies and procedures is all very well, but by themselves they do nothing. An organisation needs to check that the policies are being complied with and that the procedures are being followed to ensure that security threats are avoided or identified quickly. Therefore organisations need to carry out regular audits of their security policies and check that their employees are complying with them.

Security baselines

Security baselines are often used in organisations to define a secure starting point for a system or application. The organisation's IT security policy may, for example, define requirements such as the installation of anti-malware software and firewalls and the updating of settings. Following the written policy, system administrators can then create a baseline system from which an image can be created and deployed onto desktop computers. The baseline may cover a range of issues including the following.

- Ensuring that default 'factory settings' and 'reset' options are removed from hardware and software configurations: For example, some network devices have a default administrator password that is quoted in documentation and which is freely available. These 'factory settings' should be removed and replaced with secure, unique passwords.
- Any known backdoors should be removed: A backdoor into an IT system is a way of accessing a system by bypassing normal security requirements.
- Patches and updates: Patches and updates for hardware (firmware) and software (operating systems and security applications) should be applied and kept up to date.
- Update management: In a large computer network, the management of updates can be an issue. Rather than each individual PC in the network downloading its own updates, often the IT department will be responsible for downloading an update, testing it on a separate system to ensure that it works correctly and does not interfere with other software (sometimes called 'sandbox testing') and then managing the roll-out of the update to user machines, perhaps at night or at the weekend to minimise disruption.

Avoiding impedance of business operations

One of the issues with security policies and baselines is that a balance between security and usability needs to be maintained: that is, the rules applied for the sake of IT security should not impede the business in carrying out its day-to-day operations successfully. Examples of security rules which could cause impedance to a business's operations include the following.

- Ingress and egress of expected network traffic: Firewall rules that prevent legitimate incoming (ingress) or outgoing (egress) network traffic can create user frustration and will increase the number of IT support calls.
- Server interconnectivity: Server interconnectivity includes allowing remote access and shared folder access. Restricting this interconnectivity may make transferring data difficult and encourage people to use less secure methods such as email or USB memory sticks to transfer data.
- Time based access: Time based rules that prevent users from logging on or from accessing certain resources outside normal office hours can create problems in situations where people might need to work late.
- Remote access: Allowing users to access IT systems remotely provides many benefits such as allowing staff to work from home and to access important information while out of the office at meetings or with customers. Therefore any security policies that make it difficult to access IT systems remotely will have a negative impact on employees who work from home or work offsite.
- Allowing external access to internal servers and data interchange: Granting access to internal servers and allowing data interchange with external suppliers (banks, for example), business partners and external cloud-based solutions has many benefits and allows for efficient business processes. However, it has associated risks. For example if an external organisations' systems are hacked, this may allow hackers access, via the external links, to other companies' systems. External links therefore need careful control and monitoring.
- The impact of aggressive email filters: Email filtering is used to reduce the amount of spam received. However, aggressive spam filters can cause problems and may block legitimate emails.

- Use of different software by different individuals: Users may have their own favourite software applications, but IT security policies may specify particular applications and prevent users from installing their own software. A good example of this is internet browsers. Many people have their favourite internet browser (Firefox® and Google Chrome™ browser, for example), all of which can be downloaded for free. However, in many organisations, the downloading and installation of 'non-approved' software is not allowed.

A security baseline that is too restrictive may create a lot of user frustration and additional IT support requests. It may also encourage users to try to find ways of working around it, which may create additional security risks. Therefore a balance must be struck between security and usability for employees.

PAUSE POINT What are the network and internet usage policies at the college or school where you are studying? Identify the issues that they cover and consider the following.

- What restrictions do they place on the use of the network and why?
- Are the restrictions they place on the network reasonable?

Hint You were probably introduced to your centre's network usage policy at your course induction, but if you do not have a copy or do not know where you can access one, ask your tutor for a copy.

Extend Search on the internet for examples of network usage policies (search for 'network or internet usage policy' or 'acceptable usage policy') and compare these with your centre's policy. How and why do they differ?

Software-based protection

An organisation's IT system must have a variety of active software and hardware-based protection methods enabled. The same is true for personal IT systems, but here fewer methods will be available or needed.

Anti-virus software

IT systems need protection from malware that can attack a system from the internet or external storage devices. A variety of anti-virus and anti-malware software is available, including Windows Defender (which is built into current versions of Microsoft Windows® operating system), free anti-virus software (such as AVG AntiVirus Free Edition), and paid-for anti-virus products, which often include additional features such as password management and protection for mobile devices.

Anti-virus software generally uses two techniques to help identify malware: virus signatures and heuristics.

Virus signatures

Virus signatures are a list of code strings which can be found in individual viruses. The signatures are regularly updated because new viruses are being created all the time. The anti-virus software downloads new signatures from the software developer on a regular basis. When a virus scan is done, the software compares each file on the system and the contents of the system's memory against this signature list to see if any viruses are present.

One disadvantage of this method is that it will only detect those viruses for which signatures exist. A newly developed virus might not have its signature identified for some time after the virus is released. The time between a new virus being released and the signature to detect it being identified and downloaded onto a computer is a period in which the computer is vulnerable to so-called 'day-zero' attacks.

Heuristics

A heuristic method attempts to identify viruses by spotting the types of behaviour that viruses commonly exhibit. Heuristic methods avoid the day-zero attack vulnerability but may create more **false positives** than using virus signatures.

Dealing with identified threats

Once the anti-virus software has identified a potential threat, it will typically 'quarantine' the file to ensure that it cannot affect any other parts of the system, and it will usually notify the user. If the user is happy that the file is not important, then they can instruct the anti-virus software to permanently delete it.

Firewalls

A firewall is a software program or hardware device that filters data as it enters and leaves a network, making sure that only certain types of data can enter or leave it. (A firewall sets inbound and outbound rules for data.)

A firewall can be implemented as a software application running on an individual server or workstation. A network firewall is a hardware device that protects a whole network. Usually, a firewall creates a barrier between a LAN and the internet. Using a personal firewall, for example the one built into Windows® operating system, is essential if you are accessing the internet using public WiFi networks, such as in a café or hotel, to provide a degree of protection.

Firewalls use a number of methods to filter the data entering and leaving a network.

Packet filtering

Packet filtering involves inspecting each network data packet at the lower layers of the TCP/IP protocol, as it arrives, and applying certain rules. Any packets that do not conform to the rules are rejected. The rules can be set by a network administrator or the default rules applied by the software can be used. The rules can include source or destination IP address, source or destination port, protocol used or other settings.

Application layer filtering

Application layer filtering involves inspecting data higher up the TCP/IP protocol stack. It filters the connection on an application process basis rather than at a port level, as a packet filter does. The filter blocks any data that does not comply with the rules for the particular application.

Network address translation

Firewalls often hide the true IP addresses of devices they are protecting. This prevents external attackers from identifying the addresses of individual systems which might assist them in attacking those systems. They do this by mapping the multiple internal private IP addresses inside the LAN to the external public IP addresses used on the internet, by maintaining an address translation table.

Firewalls typically maintain a log of the network traffic they have blocked. A network or system administrator should use this log when investigating possible intrusion attempts.

Demilitarized zone and proxies

- The connection between a LAN and the internet is the place where hackers will attempt to gain access to an organisation's IT systems, so it makes sense to protect the perimeter of this network.
- Some systems, such as web and email servers, will need direct access to the internet while most will not. It is common practice to place any servers that need direct internet access in a so-called demilitarized zone (DMZ), protected by a network firewall which allows web and mail server traffic to go through.
- A second firewall protects the rest of the LAN system, blocking external requests to ports that are needed by the web and mail severs. This network configuration with a DMZ is shown in Figure 7.3.

Key terms

False positive – a false positive is when an alarm is raised by an event which is non-threatening. Anti-virus or anti-malware software and firewall systems may generate false positives when they look for the type of activity that may indicate an attack. It is impossible to prevent false positives altogether, but too many of them can be annoying and may lead to alerts being ignored, which could allow a real attack to continue without action being taken.

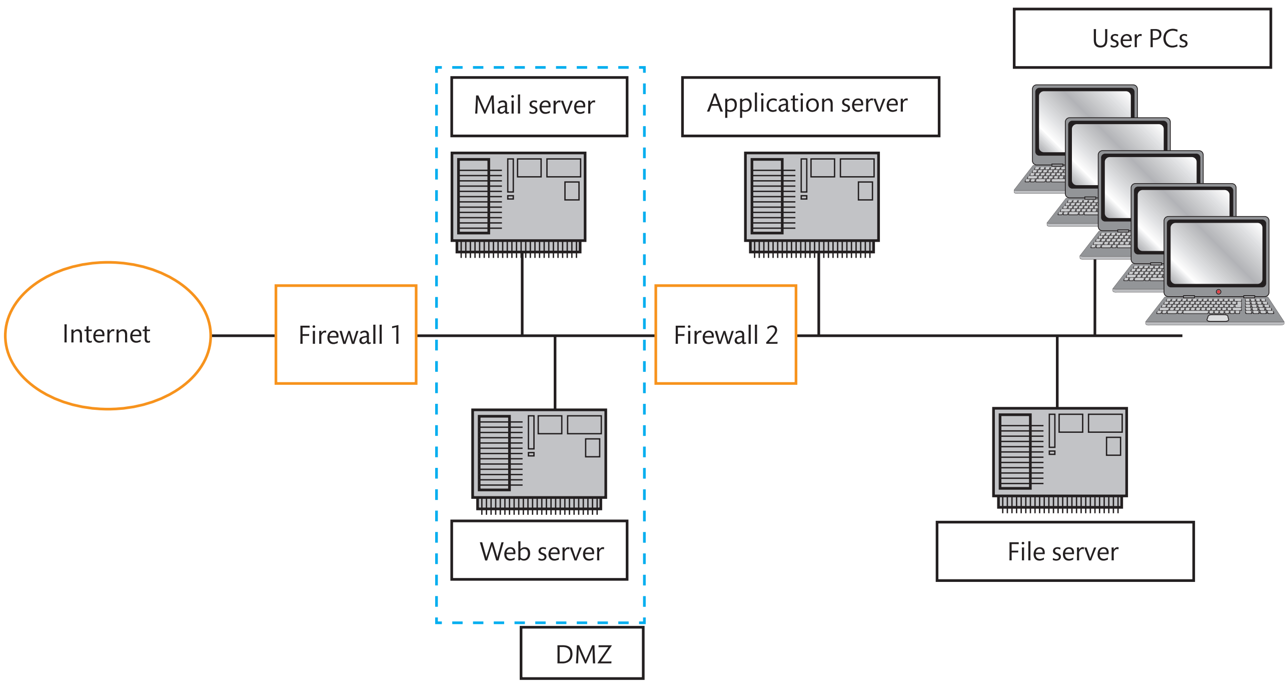


Figure 7.3: Network configuration with a DMZ

Many corporate networks also use a proxy server to forward requests for web pages from users inside the LAN. Doing so has several benefits. The proxy server can cache pages from commonly visited websites, which improves the apparent performance of the internet within the LAN as these pages do not need to be retrieved every time from the originating web server over the external internet. Proxy servers also filter internet requests and deny users access to web pages that are dangerous or inappropriate. Many third-party companies sell and update lists of websites that fall into a range of categories that companies might consider inappropriate, such as **anonymiser websites** and those for gambling and pornography. When an employee attempts to visit a filtered site, the proxy server can display a message reminding the user of the organisation's acceptable IT usage policy. Proxy servers also log which websites employees have visited.

Key term

Anonymiser websites – these provide a method for bypassing proxy server filtering.

Intrusion detection systems (IDSs)

IDSs are software applications or hardware devices that monitor a network for suspicious activity and attempt to detect and (in the case of an active IDS) block attacks on IT systems and networks. IDSs can identify malicious hacker activity that anti-virus and anti-malware software can miss. There are two types of IDS.

- Host IDS (HIDS): These protect individual servers or workstations, and are sometimes used on Internet-facing servers because email and web servers are the most vulnerable to attack.
- Network IDS (NIDS): These protect entire networks.

An IDS uses known attack signatures (similar in concept to virus signatures) of various types of suspicious activity that a hacker might carry out to gain access to the IT systems on a network. Like virus signatures, IDS attack signatures need to be updated regularly. The IDS can also look for unusual network behaviour. It does this by monitoring normal network traffic over time (a baseline) and then, if it detects network behaviour which is significantly different to the baseline, raising an alert. If suspicious activity is detected, then the IT system alerts the network administrator and may also block the activity. As most IDSs simply log events or raise alerts (perhaps by sending an automated email to the system administrator) it is essential that the system administrator regularly reviews the logs and investigates the alterations to check if they constitute a real attack, and, if they do, take appropriate action.

Research

Firewalls and IDS are a complex topic with many different technologies and configurations in use. Research the topic to find out the latest information about their features and use.

Domain management

A Windows® operating system-based client–server network is called a domain. In a large network, users do not usually log on to individual computers but are logged on to the domain via the computer they are using. Their credentials (username and password) are verified by the domain controller, which is a server computer that performs this function. This means that users do not need individual user accounts on each computer as they have just one domain account. This also makes administering users and applying security measures easier. Using a domain also allows the system administrator to restrict the devices that can attach to the domain. Before a device can join a domain, the administrator must give permission, which helps to prevent unauthorised devices from being connected to a network. (The version of Windows® operating system designed for home use cannot join a domain).

User authentication

The standard Windows® operating system log-on procedure is to enter a username and password. In a client–server network where users log on to a domain rather than a local computer, passwords are sent to the domain controller for authentication using the Kerberos protocol, which ensures that user passwords are never sent over the network unencrypted.

Research

What is Kerberos? Where does the name come from? Who developed the system and how does it work?

- Strong passwords: As with any secure system, the use of strong passwords is important to prevent others from guessing a user's passwords. Within a Windows® operating system domain, the administrator can set up policies that control the rules for acceptable passwords. Figure 7.4 shows the options that can be set for local or domain users.

The purpose of each setting for local or domain users is as follows.

- Enforce password history: When users are forced to change their passwords regularly there is a danger that they will just swap between a small number of favourite passwords, perhaps just two of them. Password history remembers a specified number of previous passwords and prevents the user from reusing those previous passwords.
- Maximum password age: Once the maximum number of days for password age has passed, the user is forced to change their password.
- Minimum password age: Allowing a user to change their password too often can also create problems. Once a user has changed their password, they must wait a specified minimum number of days before they can change it again.
- Password must meet complexity requirements: If this option is selected, then a password must meet certain requirements. For example it cannot contain the username and it must include a combination of 3 digits out of upper case, lower case, numeric, and special characters.

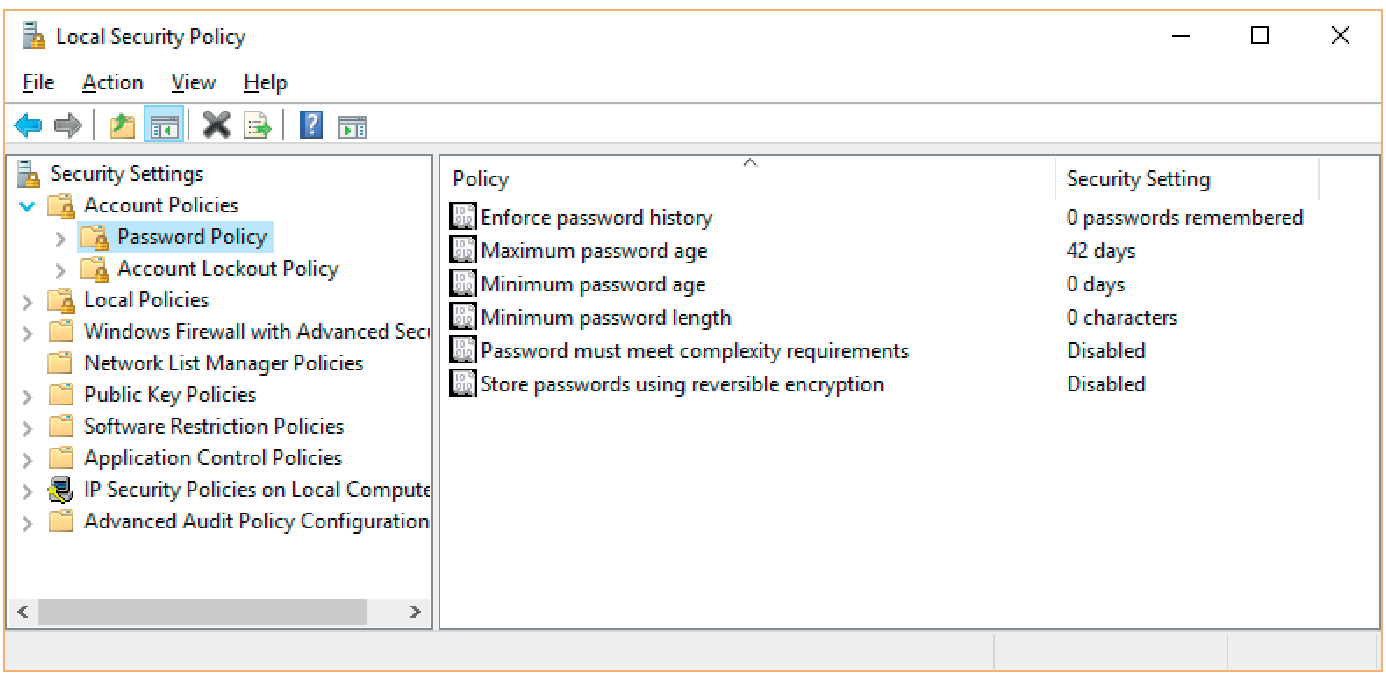


Figure 7.4: Options for local or domain users

- Store passwords using reversible encryption: This option is disabled by default and should not be enabled unless the security of the system is not important, as it allows users' passwords to be easily decrypted.

  It is also possible to restrict the time of day and days when a user can log on to the IT system, as shown in Figure 7.5.

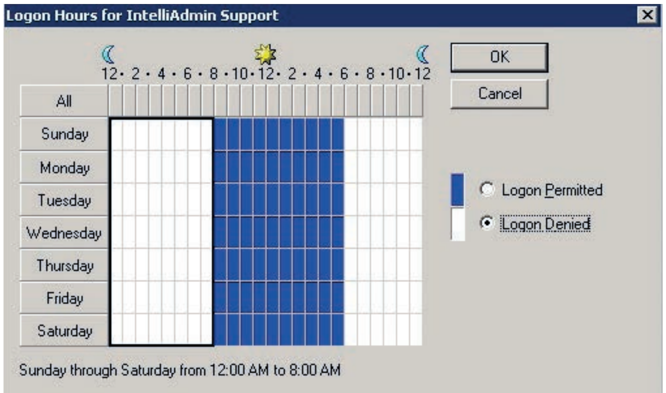


Figure 7.5: Restricting time of day and days when a user can access an IT system

Further authentication methods

Where stronger security is required beyond that provided by a password system, additional security can be provided through the use of biometric authentication (such as fingerprint readers), two-factor authentication or security tokens.

Link

Other protection methods are described in the earlier sections 'Cryptography methods' and 'Physical security'.

Discussion

Passwords are an essential part of modern life. Every time you want to access an online resource or purchase something online you need to register and create an account with a password. This could mean that you have tens, if not hundreds, of accounts all of which require a password. Discuss the best way to keep track of all these passwords with a peer or in a small group. It may not be possible or wise to use the same password for all accounts – why not?

Access controls

Network operating systems, such as Microsoft Windows® operating system and Linux, provide access controls that can be used to control which users have access to different files and folders and what type of access they have (full, write or read only access). Windows® operating system file permissions are a fairly complex topic, with slightly different permissions used for files held locally and on the servers (called NFTS file permissions) compared with those for folders shared over the network (called shared folder permissions).

The Linux operating system provides similar facilities which are known as Linux file permissions, or sometimes Linux octal file permissions. Linux file permissions use a command-driven interface and the level of access granted to a file is defined by a number between 0 and 7 (hence the name 'octal' , i.e. base 8 permissions). The meaning of the number is as follows.

Table 7.5: Linux octal file permissions

7	Read, write and execute
6	Read and write
5	Read and execute
4	Read only
3	Write and execute
2	Write only
1	Execute only
0	No access

Linux file permissions are set using a program called chmod. A three digit octal number is used to define the level of access to the file owner, the current group and everyone else.

For example, the following command would give read and write access to the file myfile to just the file owner and current group. Everyone else would have read only access to it.

```
Chmod 664 myfile
```

Windows® operating system shared folder permissions are investigated further in this following Step by step.

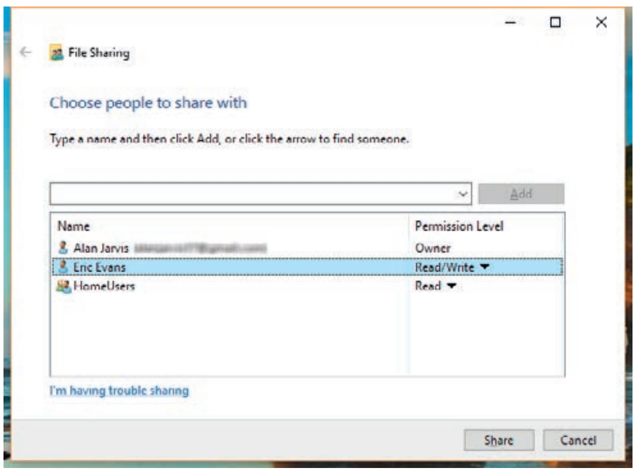
Step by step: Shared folder permissions

5 Steps

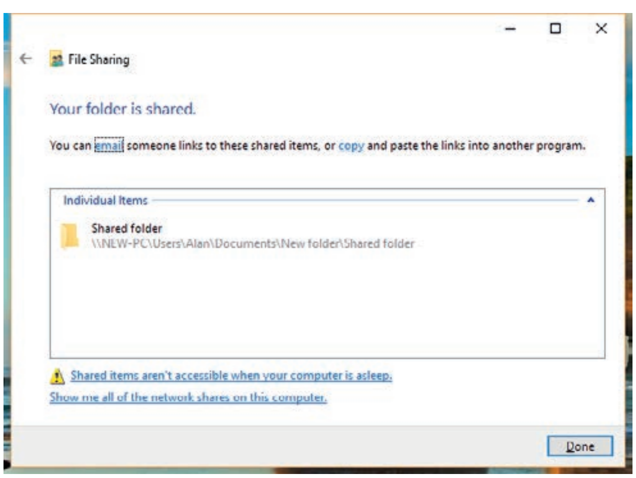
1 Imagine you want to create a shared folder but give different users different levels of access to it. First, you need to create a folder using the Windows® operating system File Explorer, then click on the folder you created and choose the **Share** tab in the menu at the top of the File Explorer window.

2 To share the folder with specific users rather than everyone, choose 'Specific people...' and the dialog box shown here will pop up.

3 If you drop down the box next to the **Add** button, you will see a list of all the users who have accounts on the computer plus 'Everyone'. Selecting **Everyone** will share your folder with everyone on the network. Click one of the accounts and then click **Add** and then that user account will be added to the box below. The default setting is that they have read only access, but you can change this to read/write by dropping down the arrow beside where it says 'Read'.



4 Click the **Share** button at the bottom and the folder is now available to everyone on the network. You will see a dialog box similar to that shown here, which gives the pathname of the shared folder.



5 Anything saved in this folder is now accessible to network users. Users with read/write access can both open and save files in the shared folder, while those with read only access can only open the files, but cannot save them back to that folder.

D Implement strategies to protect an IT system from security threats

In this section, you will look at the strategies that you can implement to protect an IT system from security threats. These include group policies, anti-malware protection, firewall configuration, wireless security and access control. You will also look at how to test and review the protection that you apply to an IT system.

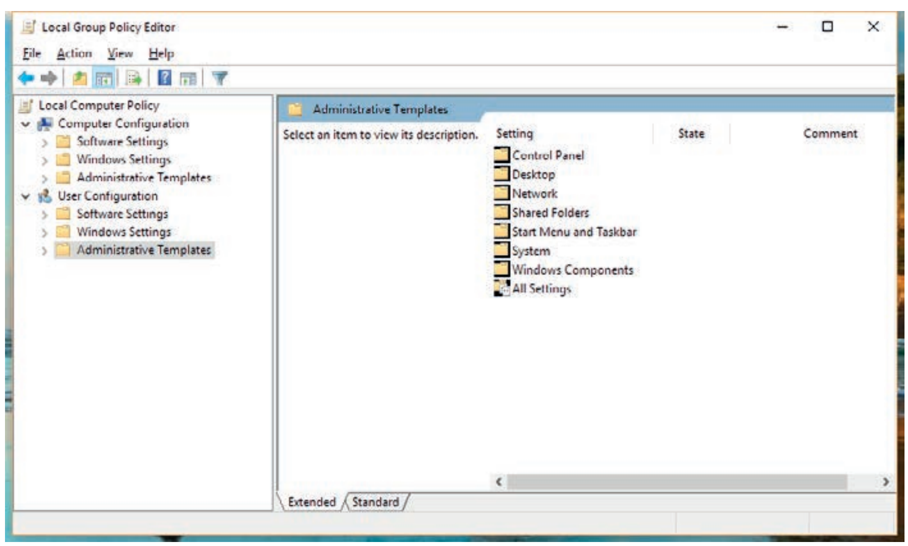
Group policies

In a large Windows® operating system client–server-based network, the system administrator can use a feature of Windows® operating system Server operating systems called 'Group Policy' to centralise the administration of the features that users or groups of users can access. Often system administrations will want to restrict users and take away features such as the ability to start programs using the run menu and the ability to make changes to the desktop. Group Policy erful tool as you only need to configure it once on the server and then it will apply to all computers attached to that domain. The system administrator should decide what restrictions should be applied to users and then use Group Policy to implement those restrictions.

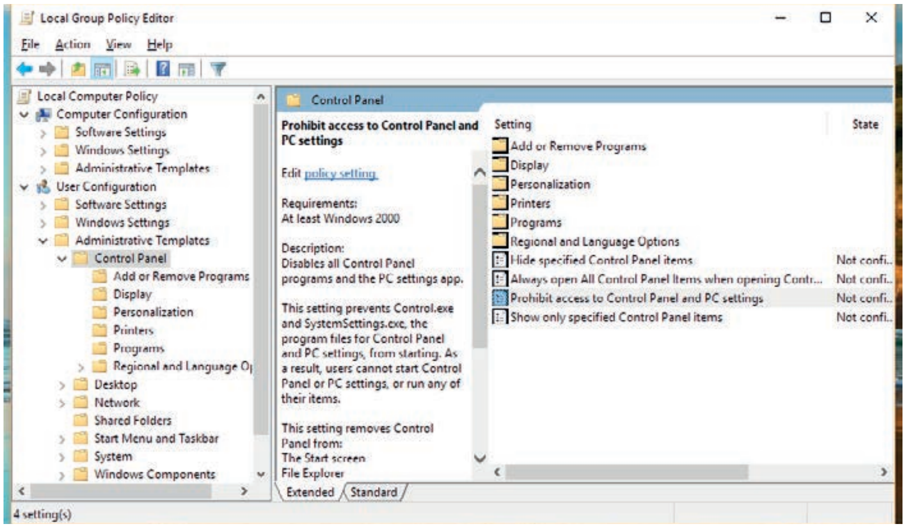
Step by step: The Group Policy editor

6 Steps

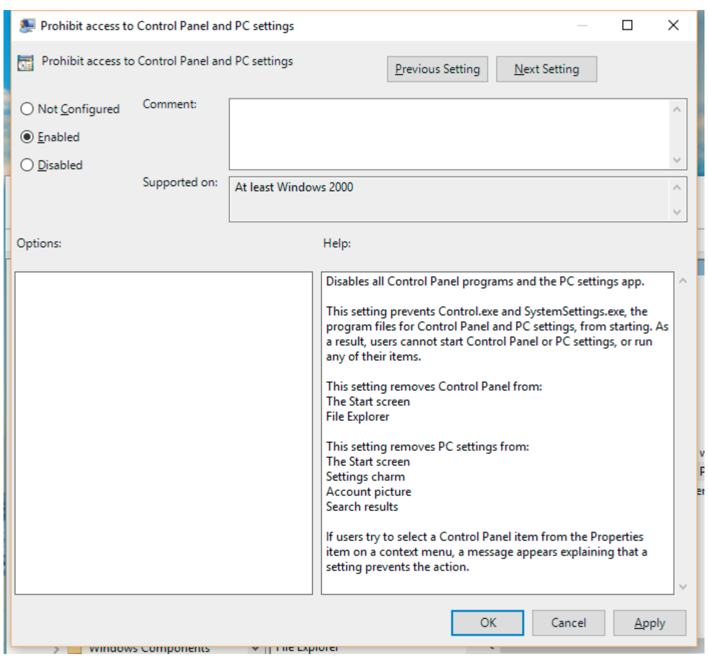
1 You can run the Local Group Policy editor on Windows® operating system 10 by using the run command and typing in 'gpedit.msc' as long as you are logged on as a user in the Administrators group. This should display the screen shown here.



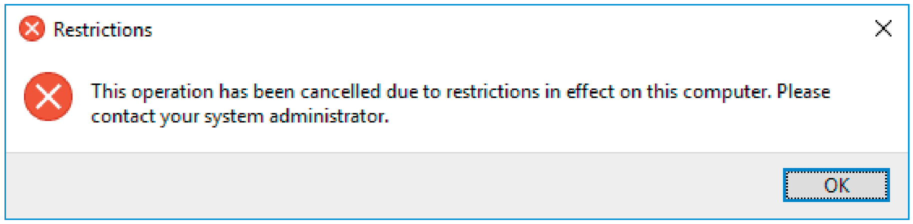
2 The Local Group Policy editor only applies settings to a single computer, but you can use similar tools on a Windows® operating system Server to apply settings to a whole domain. Click on **Administrative Templates** at the bottom and a list of templates controlling the features that a user has access to will be shown on the right. Suppose you do not want users to be able to access the Control Panel. Click on the **Control Panel** icon, and then click **Prohibit access to Control Panel and PC setting**. You will see a description of what this setting does in the centre panel.



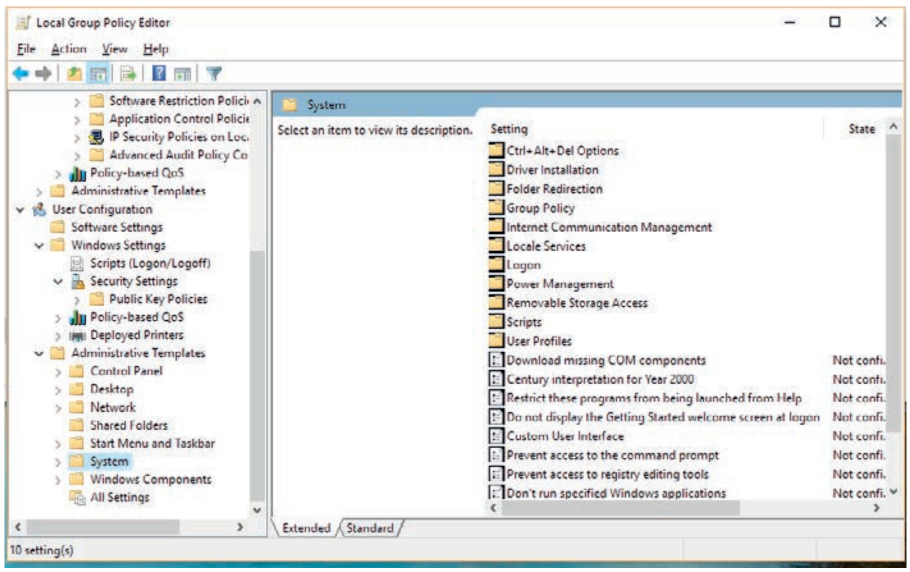
3 Double click on the option and a dialog box will pop up where you can enable this setting using the radio button.



4 Now click OK. If you now log on as a user (using an account which is not in the Administrators group) and try to open the Control Panel, you will get an error message like the one shown here.



5 There are many other settings that you can use to restrict what users can do. Many systems administrators will not want their users having access to the command prompt or to the run command. The option to disable the command prompt can be found under the System section of the **Administrative templates**.



6 The option to remove the run command is found under the **Start** menu and task bar section.

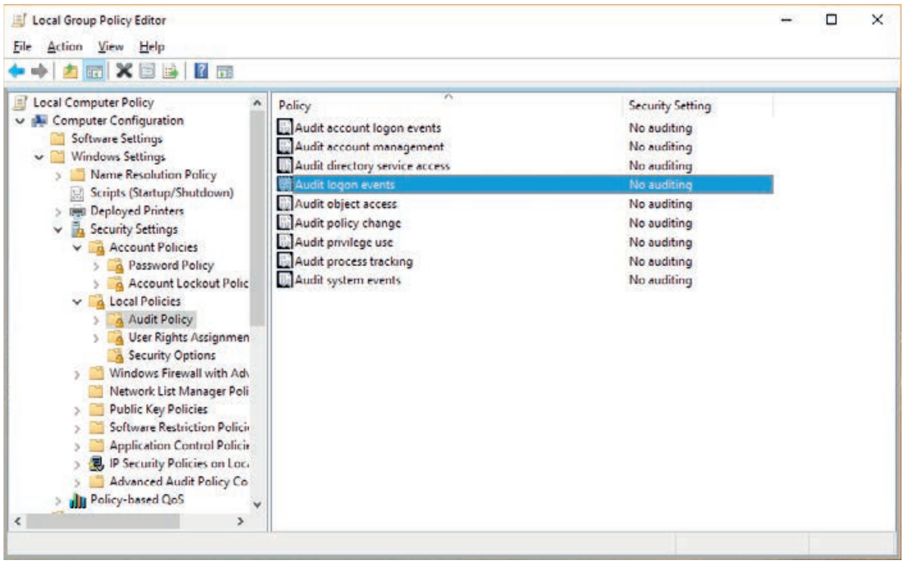
As well as using the administrative templates to restrict what users can do, the Group Policy editor also allows a system administrator to set up a range of policies that apply to the whole system. The use of Group Policy to enforce system-wide password rules was discussed earlier. This can be found under Computer Configuration>Windows Settings>Security Settings>Account Policies. As well as password policies, you can also set Account Lockout policies which allow you to lock an account after a certain number of incorrect password entries. This can be useful to defeat brute force attacks.

Other options that a system administrator might want to set are the Local Polices, which apply to security settings. These allow the system administrator to audit various events, restrict the rights that users have to carry out various tasks and a range of other options.

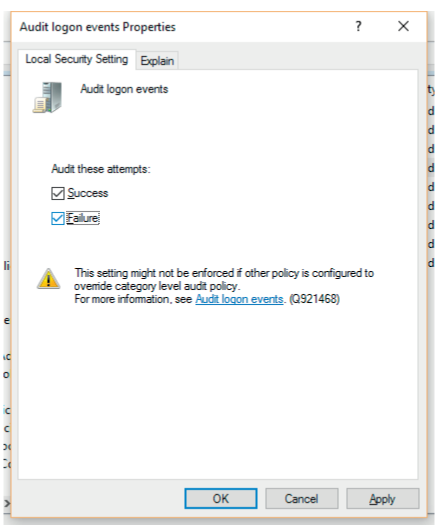
Step by step: Audit policy

3 Steps

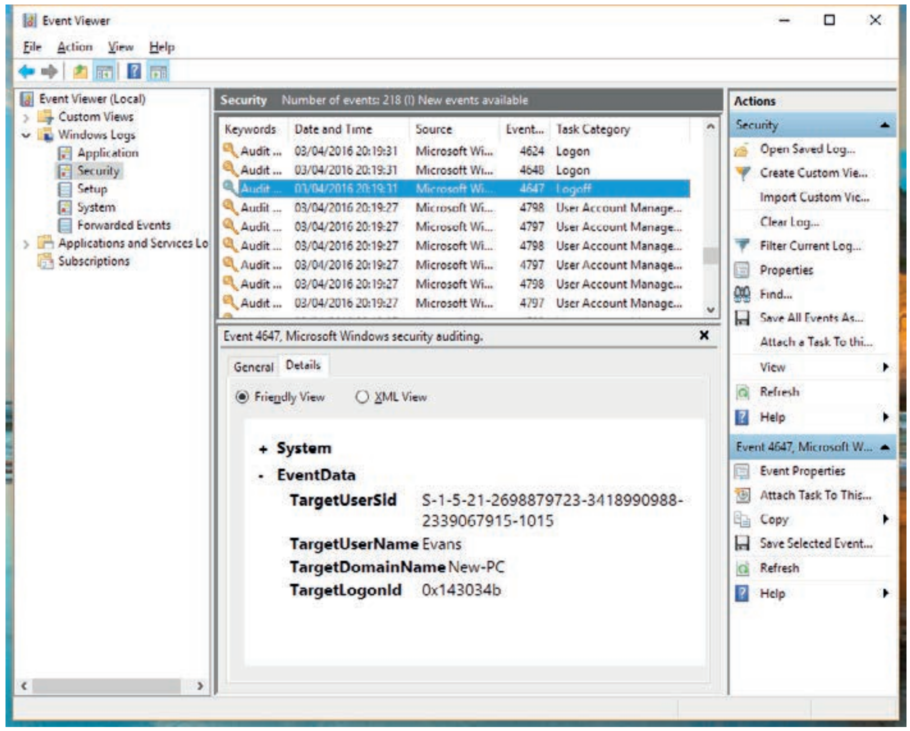
1 The audit policy allows a variety of different events to be audited in the security event log. To switch on auditing, open the Local Group Policy editor and then navigate to **Computer configuration>Windows Settings>Security Settings>Local Policies>Audit Policy**.



2 Double click on the **Audit log on events** and the dialog box shown here will appear. You can choose to audit log-on success or failure or both.



3 Auditing events causes system overheads so it is unwise to audit too many events. The Security Event log can be viewed using the Windows® operating system event viewer. Search for 'Event Viewer' and select the **Event Viewer desktop app** option. This will display the event viewer. Select **Security** on the left to see the security log, as shown here. Note that the log records a lot of information about each event, which is shown in the panel at the bottom of the display.

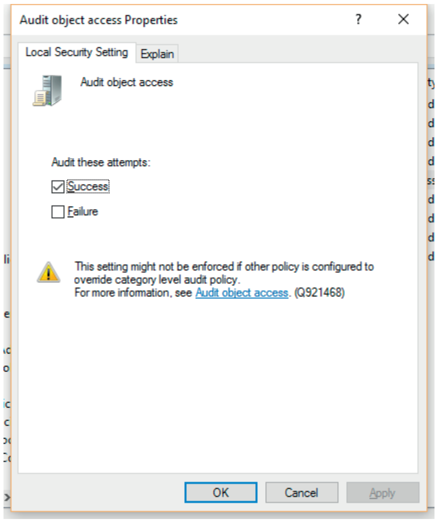


As well as auditing log-on events, you can also use Group Policy to set up auditing file access. This can be useful if you have particularly sensitive information in some files and you want to know who is accessing them and when. However, as with log-on event auditing, you should not audit too many files as it does increase the system overheads and amount of auditing information collected. With file auditing, first of all you have to switch on the ability to audit file access using the Group Policy editor, then select the files you want to audit by modifying their properties.

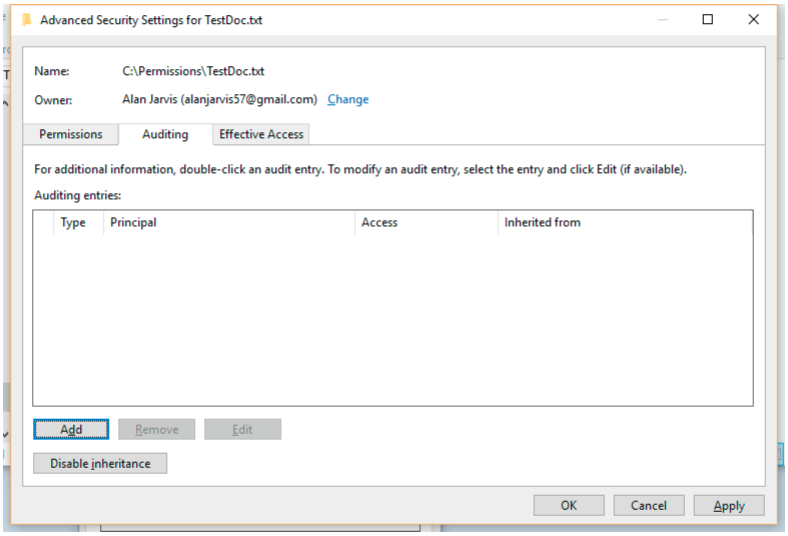
Step by step: Using audit object access

6 Steps

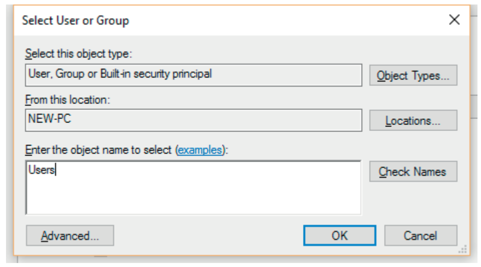
1 Open the Group Policy editor as previously and then navigate to Computer configuration>Windows settings>Security Settings>Local policies>Audit Policy. Now double click on **Audit object access** and you should see the dialog box shown here. You can audit both successful file access and failure if you wish. Auditing failure may enable you to see that someone has tried to break into files that they do not have access to. Click **OK** to close the dialog box.



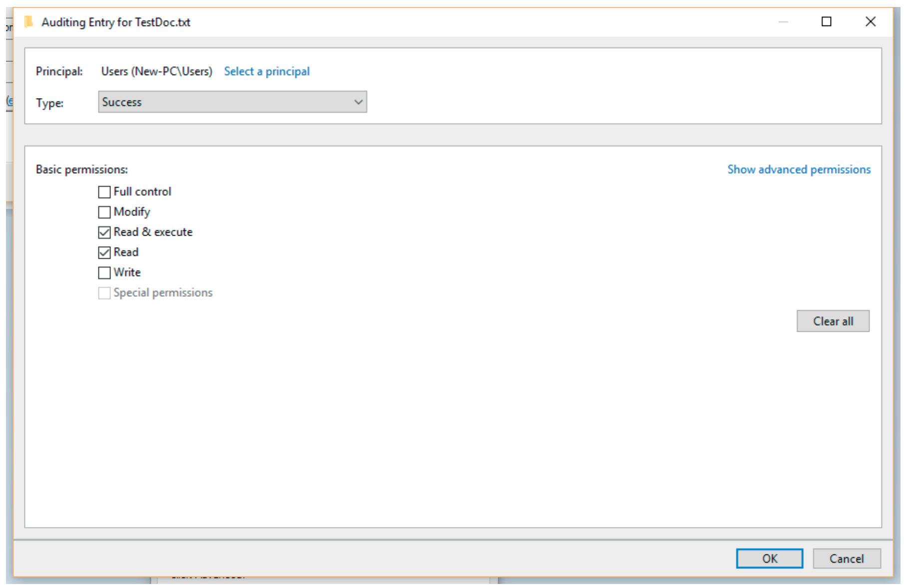
2 The next step is to select the file(s) you want to audit. Use the File Explorer to find the file, and then right click on the file and choose **Properties**. Select the **Security tab** at the top of the Properties dialog box and then click the **Advanced** button on the bottom right. This will display the Advanced Security Settings dialog box. Click the Auditing tab, and then click Continue to confirm that you have Administrator privileges. The dialog box should now look like this.



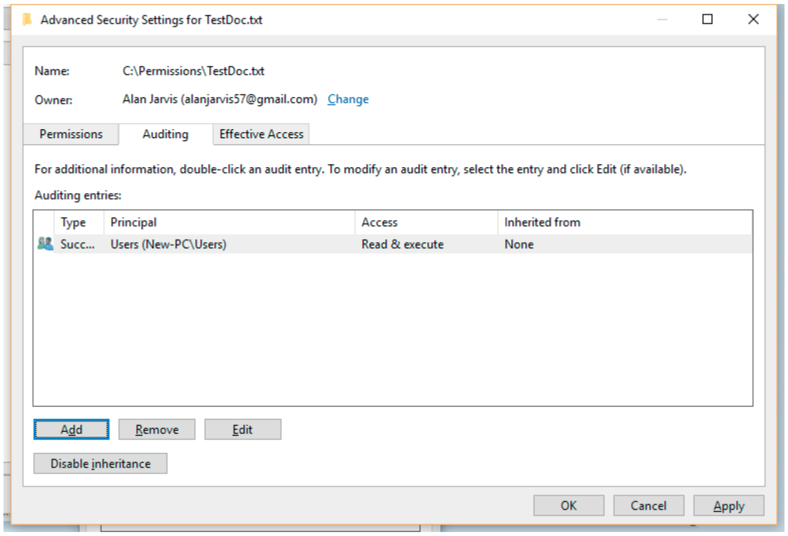
3 Click the **Add** button and the Auditing Entry dialog box will appear. Click the 'Select a principal' link at the top (i.e. who the auditing will apply to) and the Select a User or Group dialog box will appear. Enter a username or group (entering the Users group will apply auditing to all users), as shown here.



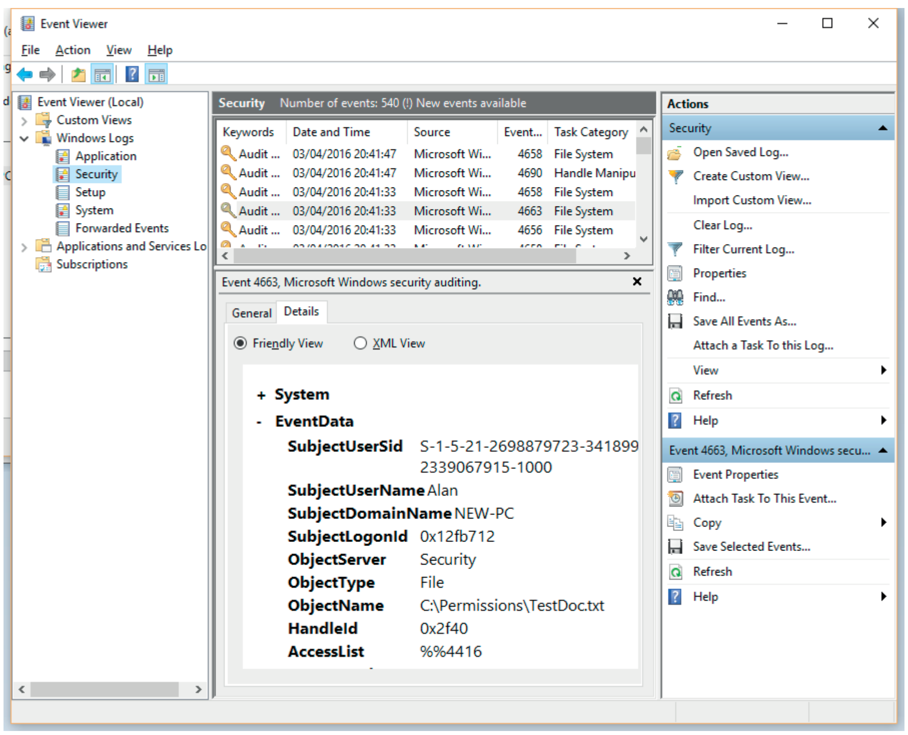
4 Now click OK to return to the Auditing Entry dialog box.



5 Click **OK** to apply these settings and you will return to the Advanced Security Setting dialog box with the auditing entry you have just created, as shown here.



6 The final step is to test that the auditing works. Open the file for which you have just made the auditing entry and then open the Windows® operating system Event Viewer, as before. Select Security on the left and you should find several File System events, as shown here. Note that simply opening the file creates several events.



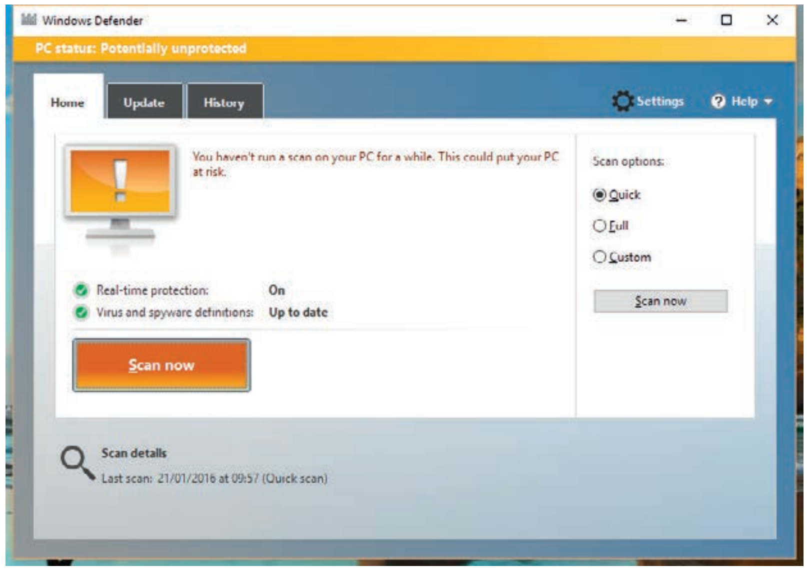
Anti-malware protection

Installing and configuring anti-malware (often called anti-virus) software is an important part of securing an IT system. Windows® operating system home editions come with basic anti-malware software called Windows Defender which is automatically installed. If third-party anti-malware software is used, then this must be installed.

Step by step: Configuring Windows Defender

3 Steps

1 You can find the Windows Defender program by searching for Defender in the Windows® operating system search box. The main screen of the Desktop App can be seen here.



2 Clicking the **Update** tab will allow you to check when the virus signature files (Windows Defender calls them Definitions) were last updated. Updates should be downloaded every few days. The History tab will let you see any suspicious files that Defender has found.

3 You can run a system scan from the Windows Defender main screen, and it is good practice to do this from time to time. You can choose to do a quick scan, which only checks those areas where malware is most likely to infect, or a full scan, which checks everywhere on the computer. A custom scan lets you check specific drives or folders, such as an external USB device.

Paid-for anti-malware products generally provide more facilities than Windows Defender. Installing a third-party malware scanner is simply a case of going to the website of the product developer and downloading the installation file and then running it. Many anti-malware software vendors provide free trials of their products, and some provide versions that are free for personal use. Currently, anti-malware protection is mainly focused on PC devices but many security experts are predicting growth in mobile-based malware targeting Apple® and Android®-based systems. Anti-malware software products (some of which are free) are available for mobile devices and their use is recommended.

Once downloaded and installed, the software should complete a scan of your system to check that there are no pre-existing malware infections. One useful facility that most third-party anti-malware software includes is a scheduled scan for new malware infections. Remembering to scan the computer regularly can be difficult and it is not a good idea to scan the computer while people are trying to use it, as scanning can slow down the system quite significantly. It is wise to scan a computer weekly, at a time when no one is using it, so setting up a scheduled scan is useful.

Step by step: Setting up a scheduled scan

6 Steps

1 In this example, a scheduled scan is set up using AVG anti-malware software – other anti-virus software would be set up in a similar way. First open the AVG user interface by clicking the icon in the task bar.

2 Then click the gear icon next to the Scan now button to open the Scan options dialog.

3 Next click the **Manage scheduled scans** button which will open the scheduled scans dialog. Click the **Add scan schedule** button at the bottom left and you will see the Schedules scan options.

4 Here a scan has been scheduled for every Sunday at 10pm. Click the **Save** button to confirm the new schedule (you may be asked by User Account Control if you want to allow this change). The scheduled scans dialog should now show that a scan is scheduled for the following Sunday at 10pm.

5 If, during a scan, AVG finds a file that it considers to be malware, then you can view a report at the end of the scan which gives details of the file it has found.

6 Clicking on the 'More info' link will take you to a page on the AVG website that describes the malware it has found in more detail. AVG also provide a number of configurable options that allow you to adjust the way in which scans are done and when updates are downloaded.

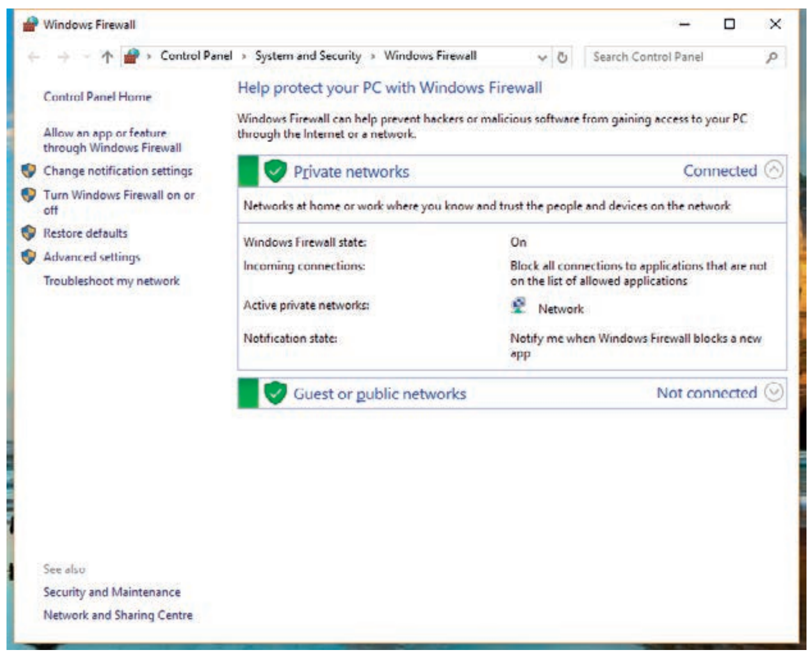
Firewall configuration

Windows® operating system also includes a basic firewall that is automatically configured and switched on when Windows® operating system is installed. The firewall enables you to allow or block various applications and set rules for both inbound and outbound data.

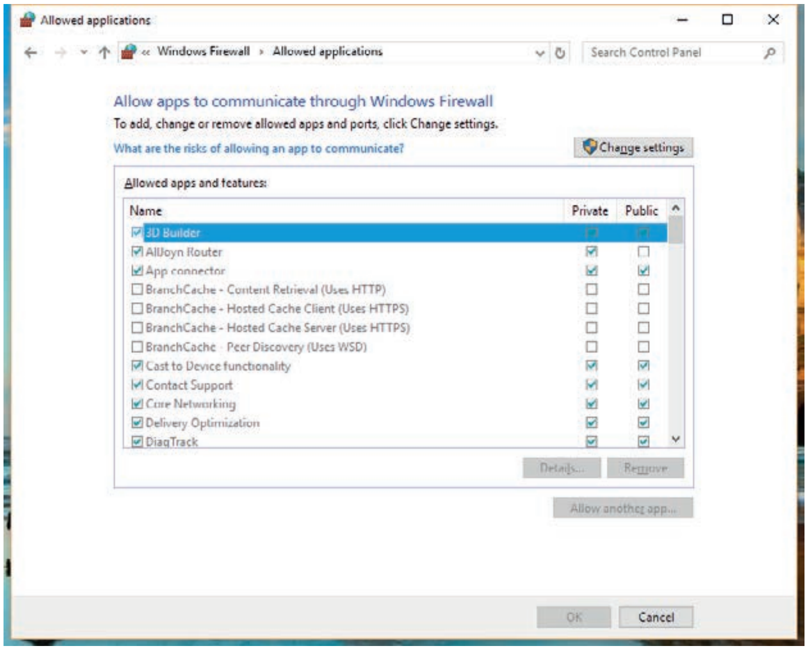
Step by step: Configuring the Windows® operating system firewall

8 Steps

1 To open the Windows® operating system Firewall user interface, type firewall in the Windows® operating system search box and click on the Windows® operating system Firewall – control panel option. This should display the screen shown below. This screen shows the status of the firewall. Windows® operating system Firewall allows you to configure different settings depending on whether you are connected to a private network (e.g. at home) or a public network (e.g. a public WiFi hotspot). The main screen should show that the Windows® operating system Firewall is on. Connections that are not on the allowed list are blocked and notifications are turned on.

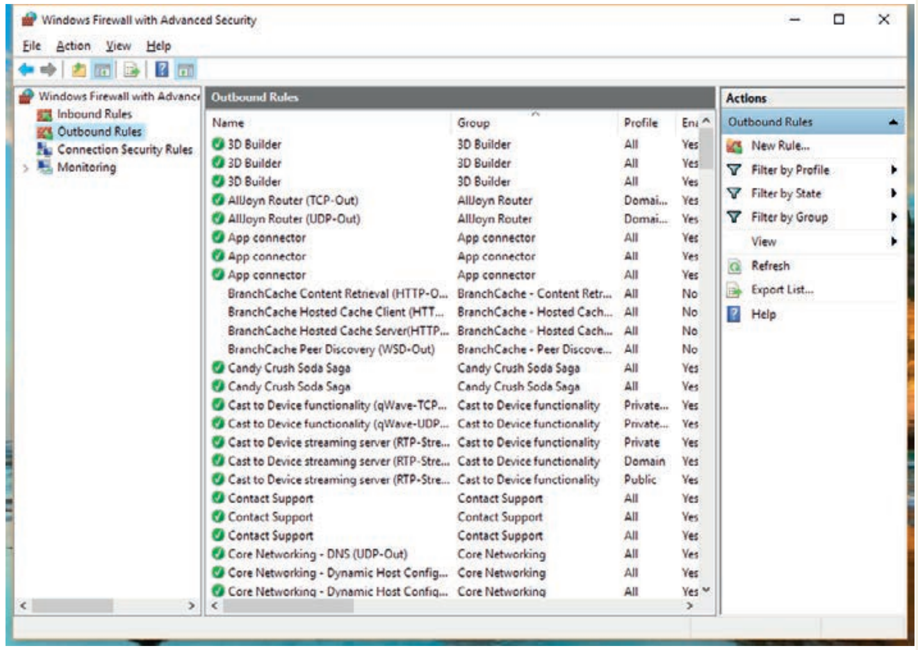


2 If a network application is not working correctly because the firewall is blocking it, you may need to do the following to allow it through the firewall. Click the 'Allow an app or feature through the Windows® operating system Firewall' link on the left. This will display a window similar to that shown here.

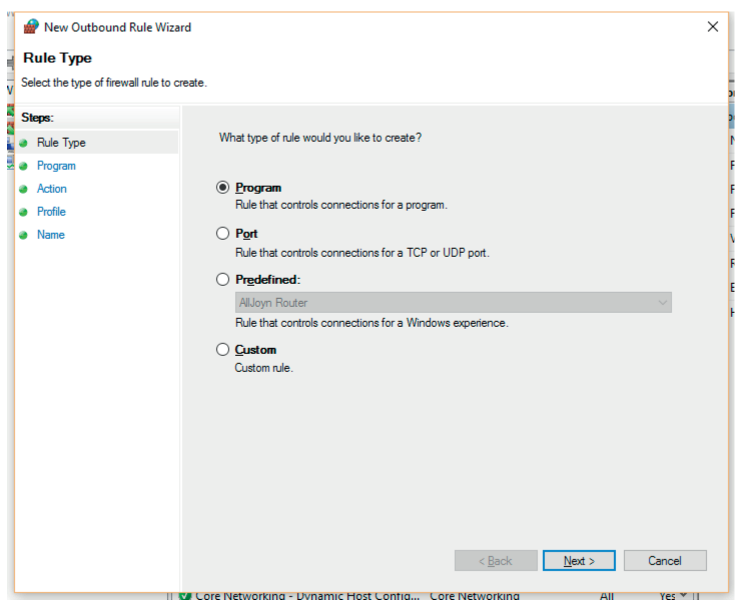


3 Click the change settings button (enter the administrator password if prompted) and scroll down the list of applications to find the one you want to allow. Then click the Private and/or Public check boxes to allow access on Public and/or Private networks. Click **OK** to complete the process.

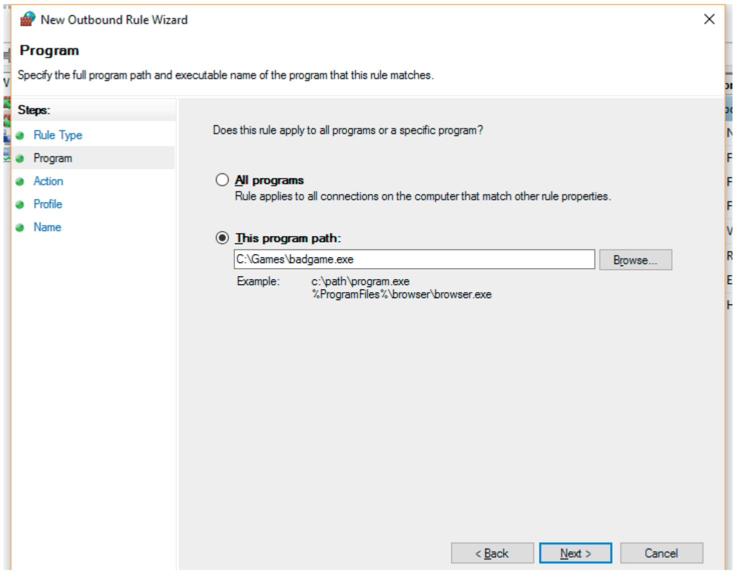
4 Windows® operating system Firewall can also be configured to block specific port, protocol or IP addresses. Click the **Advanced Settings** link on the left of the main firewall screen and you will see the Windows® operating system Firewall with Advanced Security window as shown here.



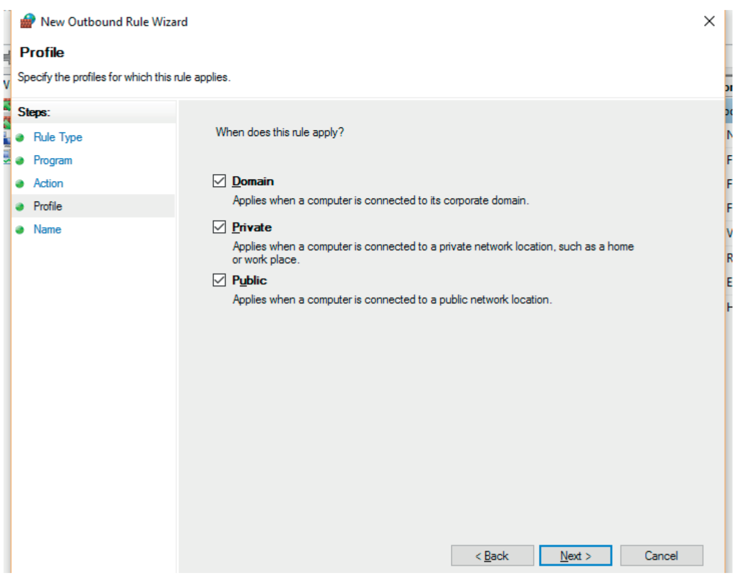
5 Suppose you want to stop anyone playing a game called 'badgame.exe' which connects to the internet. Choose Outbound rules on the left. These are the rules that apply to data coming from the computer to the internet. In the central pane, you can see all the existing outbound rules. Unless you know what you are doing it is not wise to change any of these. Click **New Rule** under the Actions pane on the right. This will start the New Outbound Rule Wizard as shown here.



6 The default setting is for a 'Program' rule, which is what is required, so just click **Next**. In this step, you need to enter the name of the program file you wish to block, as shown here.



7 Click **Next**. The next step will ask you what action you want to take, which, in this case, is the default 'Block the connection', so just click **Next**. On the Profile step, you need to decide when you want the rule to apply, that is, when the computer is connected to a domain, a private network or a public network, as shown here.



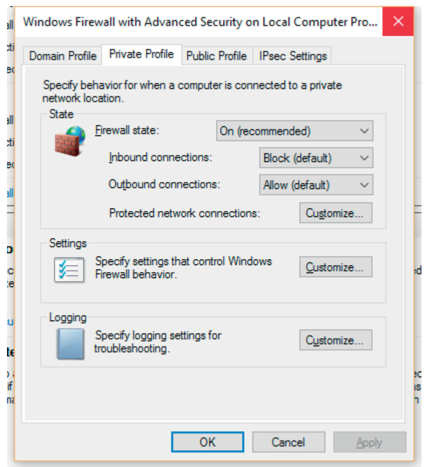
8 You may want the rule applied only when connected to a public network such as a WiFi hotspot in a café or railway station, but, for this exercise, you can leave all the options selected. Click **Next**. On this step you give the rule a name and a description, which can be helpful if you or someone else does not know or remember what the rule is for. Click the **Finish** button to complete the rule, which is now added to the list of outbound rules.

The Windows® operating system Firewall can log both successful and unsuccessful network connections. This is useful for trouble shooting or if you suspect hackers may be gaining access to your system. The log collects a lot of data so it is unnecessary for it to log all the time.

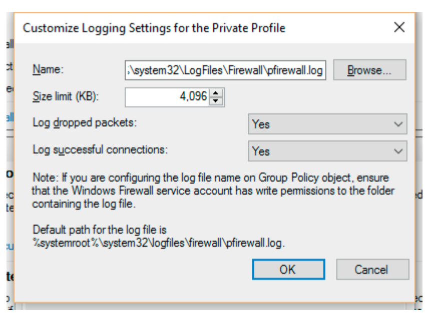
Step by step: Enable firewall logging

5 Steps

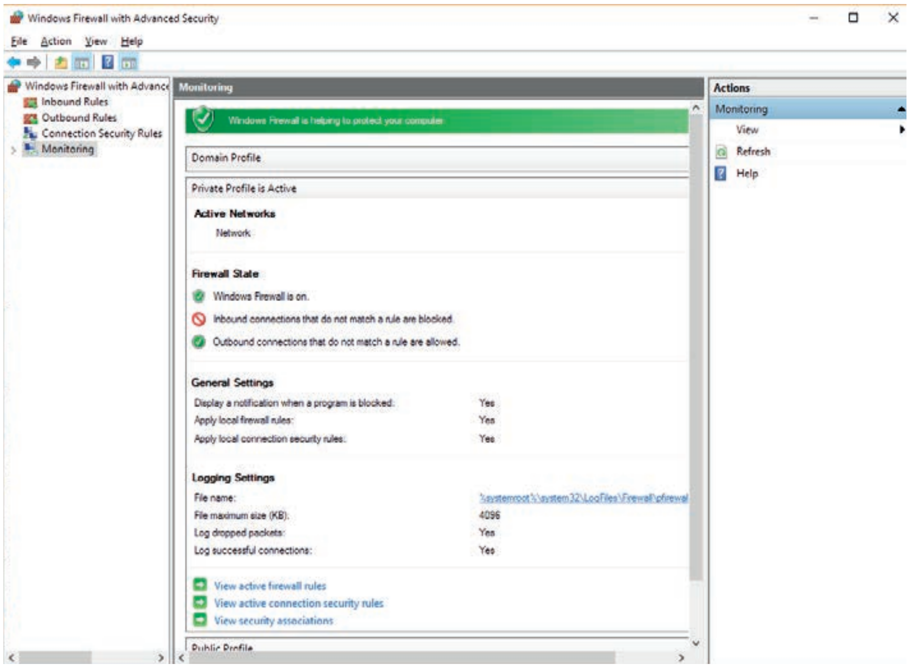
1 First, open the Firewall with Advanced Security window, make sure that in the left-hand pane you have 'Windows® operating system Firewall with Advanced Security' selected (i.e. not any of the rules) and then, on the right-hand pane, click **Properties**. This will open the properties dialog box, as shown here.



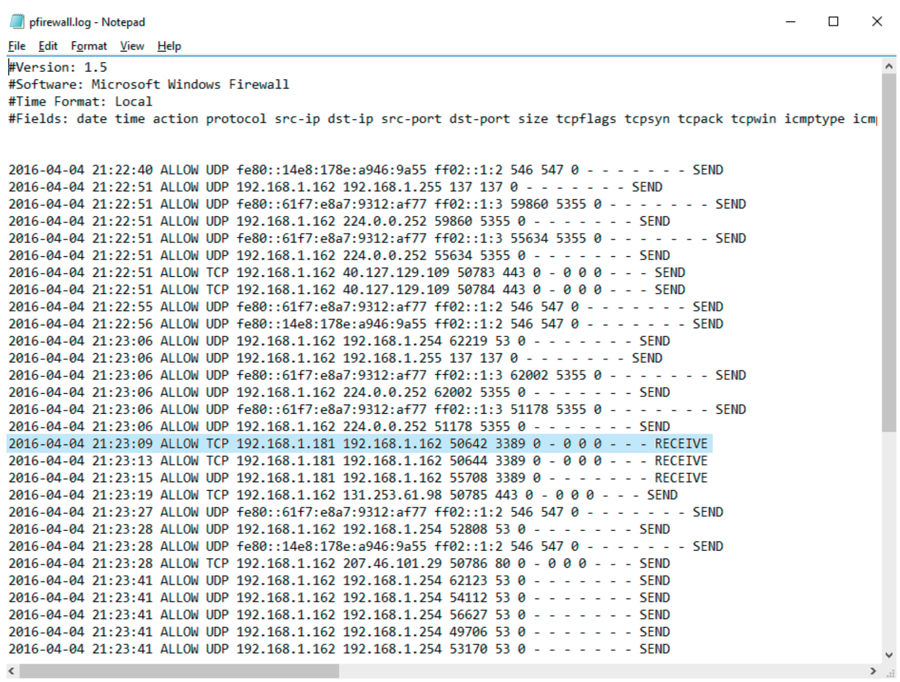
2 Select the **Private profile** tab at the top of the dialog box (assuming that your current network is recognised by Windows® operating system as a Private network). Click the **Customize** button in the logging section. You will then see the Customize Logging Settings dialog box, as shown here. Leave the file name for the log file and the size limit as they are and set **Log dropped packets** and **Successful connections** to **Yes**.



3 Click **OK**, and then **OK** on the properties dialog. In this example, a remote desktop session was opened with the computer and then the log file was viewed. To view the file, click the link at the top left of the Windows® operating system Firewall with Advanced Security main window. Scroll down the central pane of the window and click the link at the bottom labelled 'Monitoring'. This will display the monitoring settings, as shown here.



4 Towards the bottom of the pane under the Logging Settings section, there is a link to the log file. Click this link and the log file will open in Notepad, as shown here.


pfirewall.log - Notepad

File Edit Format View Help

```
#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmptype icm


2016-04-04 21:22:40 ALLOW UDP fe80::14e8:178e:a946:9a55 ff02::1:2 546 547 0 - - - - - - - SEND
2016-04-04 21:22:51 ALLOW UDP 192.168.1.162 192.168.1.255 137 137 0 - - - - - - - SEND
2016-04-04 21:22:51 ALLOW UDP fe80::61f7:e8a7:9312:af77 ff02::1:3 59860 5355 0 - - - - - - - SEND
2016-04-04 21:22:51 ALLOW UDP 192.168.1.162 224.0.0.252 59860 5355 0 - - - - - - - SEND
2016-04-04 21:22:51 ALLOW UDP fe80::61f7:e8a7:9312:af77 ff02::1:3 55634 5355 0 - - - - - - - SEND
2016-04-04 21:22:51 ALLOW UDP 192.168.1.162 224.0.0.252 55634 5355 0 - - - - - - - SEND
2016-04-04 21:22:51 ALLOW TCP 192.168.1.162 40.127.129.109 50783 443 0 - 0 0 0 - - - SEND
2016-04-04 21:22:51 ALLOW TCP 192.168.1.162 40.127.129.109 50784 443 0 - 0 0 0 - - - SEND
2016-04-04 21:22:55 ALLOW UDP fe80::61f7:e8a7:9312:af77 ff02::1:2 546 547 0 - - - - - - - SEND
2016-04-04 21:22:56 ALLOW UDP fe80::14e8:178e:a946:9a55 ff02::1:2 546 547 0 - - - - - - - SEND
2016-04-04 21:23:06 ALLOW UDP 192.168.1.162 192.168.1.254 62219 53 0 - - - - - - - SEND
2016-04-04 21:23:06 ALLOW UDP 192.168.1.162 192.168.1.255 137 137 0 - - - - - - - SEND
2016-04-04 21:23:06 ALLOW UDP fe80::61f7:e8a7:9312:af77 ff02::1:3 62002 5355 0 - - - - - - - SEND
2016-04-04 21:23:06 ALLOW UDP 192.168.1.162 224.0.0.252 62002 5355 0 - - - - - - - SEND
2016-04-04 21:23:06 ALLOW UDP fe80::61f7:e8a7:9312:af77 ff02::1:3 51178 5355 0 - - - - - - - SEND
2016-04-04 21:23:06 ALLOW UDP 192.168.1.162 224.0.0.252 51178 5355 0 - - - - - - - SEND
2016-04-04 21:23:09 ALLOW TCP 192.168.1.181 192.168.1.162 50642 3389 0 - 0 0 0 - - - RECEIVE
2016-04-04 21:23:13 ALLOW TCP 192.168.1.181 192.168.1.162 50644 3389 0 - 0 0 0 - - - RECEIVE
2016-04-04 21:23:15 ALLOW UDP 192.168.1.181 192.168.1.162 55708 3389 0 - - - - - - - RECEIVE
2016-04-04 21:23:19 ALLOW TCP 192.168.1.162 131.253.61.98 50785 443 0 - 0 0 0 - - - SEND
2016-04-04 21:23:27 ALLOW UDP fe80::61f7:e8a7:9312:af77 ff02::1:2 546 547 0 - - - - - - - SEND
2016-04-04 21:23:28 ALLOW UDP 192.168.1.162 192.168.1.254 52808 53 0 - - - - - - - SEND
2016-04-04 21:23:28 ALLOW UDP fe80::14e8:178e:a946:9a55 ff02::1:2 546 547 0 - - - - - - - SEND
2016-04-04 21:23:28 ALLOW TCP 192.168.1.162 207.46.101.29 50786 80 0 - 0 0 0 - - - SEND
2016-04-04 21:23:41 ALLOW UDP 192.168.1.162 192.168.1.254 62123 53 0 - - - - - - - SEND
2016-04-04 21:23:41 ALLOW UDP 192.168.1.162 192.168.1.254 54112 53 0 - - - - - - - SEND
2016-04-04 21:23:41 ALLOW UDP 192.168.1.162 192.168.1.254 56627 53 0 - - - - - - - SEND
2016-04-04 21:23:41 ALLOW UDP 192.168.1.162 192.168.1.254 49706 53 0 - - - - - - - SEND
2016-04-04 21:23:41 ALLOW UDP 192.168.1.162 192.168.1.254 53170 53 0 - - - - - - - SEND
```

5 The highlighted line in the log shows the following:

2016-04-04 21:23:09	the date and time of the log entry
ALLOW TCP	The firewall allowed a connection using the TCP protocol
192.168.1.181	The source IP address on the connection
192.168.1.162	The destination IP address of the connection
50642	The source port
3389	The destination port (Remote desktop connections use port 3389)
0	The size of the data transferred
- 000----	The TCP flags and other data
Send	This was a connection that sent data

The log data can be useful but it sometimes needs interpretation. The following information can be obtained from looking at the log.

- **Source and destination IP addresses:** This tells you which computers were sending/receiving data.
- **Source and destination ports:** This tells you which applications were sending/receiving data. If you do not recognise a port number, try the list of port numbers here: https://technet.microsoft.com/en-us/library/cc959828.aspx
- **TCP flags and other data**: This information can only be interpreted with a detailed knowledge of how the TCP and UDP protocols work.

Wireless security

Most modern WiFi routers/access points come with effective security settings preconfigured. However, you can check and adjust them if needed. The exact method you need to use will depend on the type of router you have and you will need to search online for the manual or instructions for your particular router. Most routers provide a web-based user interface that you access from your computer via a browser.

Step by step: Configuring a wireless router

3 Steps

1 This example shows how to check and configure the settings on a BT Home Hub router. To access a BT Home Hub, just enter the URL http://bthomehub.home from a browser and you will see the main menu screen.

2 The main screen shows information about the status of the hub and the devices connected to it. To change the settings, click the **Settings** link at the top. You will need the admin password to access these settings, which is printed on the card at the back of the router. The **Advanced settings** link allows you to change some important settings, such as the type of security used and the password. Take care when making any changes to these settings, as you could reduce the strength of the security on you network.

3 These settings show that WPA and WPA2 wireless encryption are turned on with a reasonably strong password set (see the section Applications of cryptography). There are other settings that can be adjusted to improve the security of the hub. Under the Settings link, there is **BT Access Control tab** which allows you to control the times of day that all or certain devices have access to the internet.

Access control

For a large organisation, designing an effective regime of user groups and folder permissions can be quite complex. Ideally, based on the information security principles, you want to give users only the minimum amount of access to folders that they require to do their job. In reality, a regime that is too restrictive risks creating a lot of support requests because users are unable to access files they need to.

Link

For more about information security principles, see the section 'Information security'.

In a medium or large organisation that uses a Windows® operating system-based PC network, the system would normally be configured to use a client–server domain-based set-up, with one or more server computers running a Windows® operating system Server operating system. Users are set up centrally on the server that is configured as the domain controller and, to simplify the management of users, they would normally be placed in groups which define their access to files, folders and other resources such as printers. Windows® operating system has a number of built-in groups, of which the Administrator group is the most powerful. Users should only be added to the Administrator group when they need to have full control over all the system, and members of the Administrator group should have the strongest passwords, since they represent a considerable security risk.

Step by step: Creating users and groups

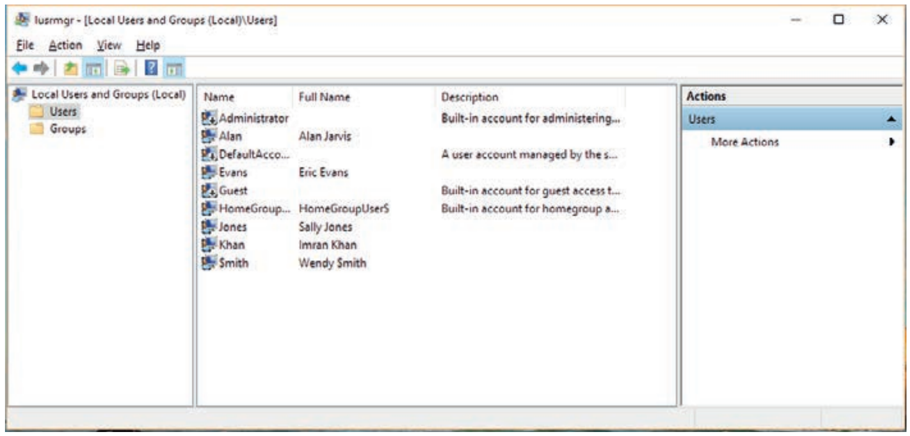
13 Steps

This example shows how groups and users can be set up on a local computer running Windows® operating system 10 rather than a server computer. However, the procedures are very similar to those that would be used on a server. Three user groups – for staff, managers and directors – are required. The users in each group are as follows (in reality, of course, there would be many more users).

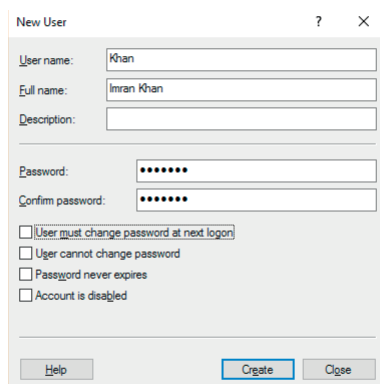
Eric Evans – Staff
Sally Jones – Staff
Imran Khan – Manager
Wendy Smith – Director

Step 1

1 To create the users and groups, you use the local user manager program, which is very similar to the Windows® operating system Server user manager program. To run the program, type **Run** into the Windows® operating system search box and then select the Run desktop app. Type the name of the local user manager program **lusrmgr.msc** into the Open box and click **OK**. The program should then run, as shown here.



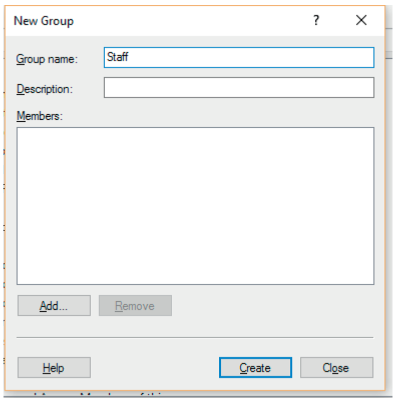
2 To create new users, click **More Actions** on the left and then choose **New User...** You will then see the dialog box shown here.



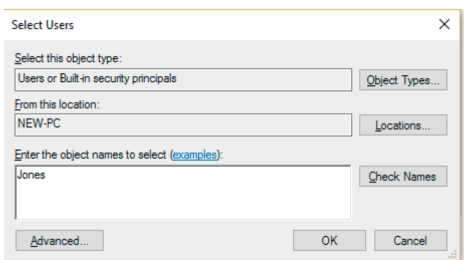
3 Enter each of the users listed above, set their password to something simple like 'welcome' and turn off the 'User must change password at next logon' option. This is just to make experimentation easier – these options should not be used for real users. Click **Create** to make each account and then **Close** when you have finished.

Step 2

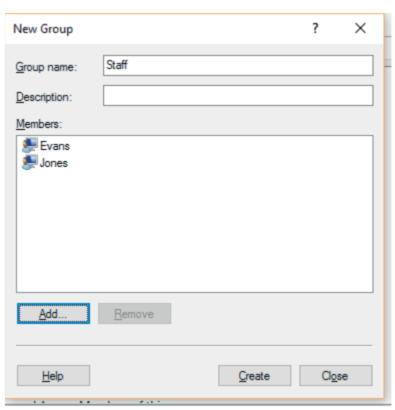
1 Next, we will create the groups and add the users to the correct groups. Windows® operating system comes with a large number of built-in groups, which you can see when you click on the **Group** link on the right-hand side of the Local User Manager program. To create new groups, click on **More Actions** on the right, then choose **New group**. The New group dialog box is opened, as shown below. Create the Staff group by typing the name in the Group name box at the top of the dialog box (there is no need to add a description).



2 To add a group member, click the **Add** button then enter the first username, as shown here.

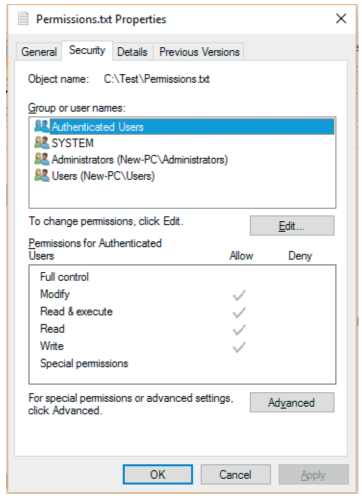


3 Click **OK** and this username will be added to the Staff group, and then do the same for the other user (Evans). The New Group dialog box should now look like this. Click **Create** to create the new group. Now do the same for the Manager and Director groups, adding the correct user to each one.



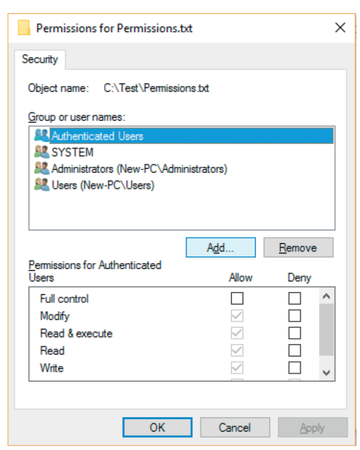
Step 3

1 Next, you will see how file permissions can be used to control the access that the different groups have to files. First, you will need to create a new folder in the root of the C: drive. Open the Windows® operating system File Explorer and navigate to the root of the C: drive. Create a simple notepad file or other document in the folder you have created. Then right click on the file and choose **Properties** to open the properties dialog box for the file. Click on the **Security** tab, which should look something like this.

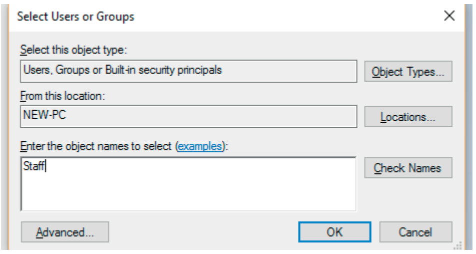


2 The top part of the security dialog box shows the groups or individual users who already have access to the file. If you click on a particular user or group, you can see, in the lower part of the dialog box, the permissions that the selected user or group has to the file. Users in the Administrators group have access to the file, as does anyone in the Users group, in which all local user accounts are automatically. This means that, at the moment, any of the other users you have created will have access to the file, as they are in the Users group. You can log in as one of the users you created to check that this is the case.

3 Suppose we did not want anyone in the Staff group to access this file, but want the Managers group to have read access and the Directors to have read and write access (they have this already so no changes are needed to achieve this). To change the permission on the file, you need to click the Edit button on the properties dialog box, which will then display another dialog box, as shown here.

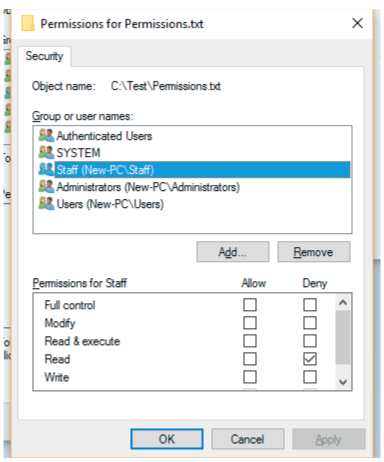


4 Now click the **Add** button on this dialog box to add the Staff group. Then you will see the Select users or groups dialog box. Enter the name of the Staff group, as shown here.

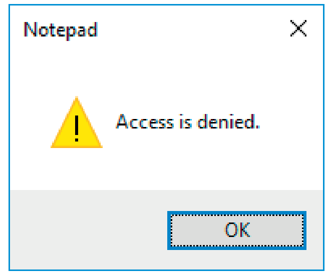


5 Now click **OK** and you will be returned to the previous dialog box, with the Staff group added to the top of the list of Groups or Users. Click on the Staff group name to see the current permissions in the lower part of the dialog box. It will look as if the Staff group only has read and read and execute permissions, but the users are also members of other groups (the Users group, for example) which gives the users other permissions. This is one of the issues that can make understanding permissions quite complex. You may have noticed that each permission can be either allow or deny. Normally, you use allow permissions, and these are cumulative – in other words, if any group that a user is a member of has, for example, write allow permission, then that user also gets write permission for the file. However, deny permissions work the opposite way around, so that if any group the user is a member of has write deny permission, then the user cannot write to the file even if other groups the user is a member of do have write allow permission.

6 Therefore, to prevent the Staff group having access to this file, we need to set the read deny permission to on. We do not need to set the other deny permission to on because, if they cannot read a file, then obviously they cannot write to it. This setting is shown here.



7 You should now find that if you log on as one of the users who is a member of the Staff group (Eric Evans), for example, and try to open that file, you will not be able to and will get the message shown here.



File and folder permissions are a complex issue and, to understand it well, you will need to practise and test your understanding of how it works. There are lots of Microsoft® training materials and books which cover the topic in detail. A poorly understood or complex set-up of groups and file permissions can cause an organisation a lot of wasted time, frustrated IT users and a lot of IT support calls.

Link

For more about access control, specifically on defining password policies, including length, complexity, age and reuse for desktop and server computers, see the section 'User authentication', in 'Software-based protection'.

There are a number of related issues such as the following.

- Ensuring that only applications whose updates (including binary executable files) are signed using a hash can be 'white listed' as trusted files.
- Ensuring that sensitive files, such as log files, are hidden and protected from unauthorised access. Typically, this would involve using the permissions settings, as described above.

Testing and reviewing protection

Setting up the required protection on an IT system is the first step to ensuring that a system is secure, but you also need to test that the protection works as it should and, from time to time, you should retest it to check that it still provides sufficient protection.

As with software development, the standard way to test a system is to create a test plan. The test plan identifies what you are going to test and what outcomes you expect before you start doing the tests. A suggested format for a security test plan is shown in Table 7.6.

Table 7.6: Security protection test plan

Test number	Risk targeted	Test description	Expected outcome	Actual outcome
1	Virus infection	Check definitions are updated	Definitions are the latest version	
2	Virus infection	Run virus scan	Scan finds no virus or resolves minor infection	
3	External hacking	Check firewall – port 80	Port is open	
4	External hacking	Check firewall – port	Port is open	
5	External hacking	Check firewall – port	Port is open	
6	External hacking	Check firewall – all other ports	Ports are closed	
7	Etc.			

A test plan like this can be used to systematically test that the firewall deals correctly with both allowed and blocked entry points and therefore blocks unauthorised traffic and allows legitimate traffic through.

Manual testing using a test plan can be useful, but it is only really suitable for a single IT system (individual computer). In a situation where a whole network of IT systems needs to be tested, a different approach is needed.

Network testing tools

An alternative to using the traditional method of manual testing is to use network testing tools. These tools can make testing a large number of risks on the many different IT systems (computers) in a medium or large network much easier.

Vulnerability scanners

Vulnerability scanners are a specialised form of software that uses a database of known security vulnerabilities to test whether a system is protected against them. The tests the software can do include identifying issues such as the following.

- **Open ports:** The scanner software can check for systems on the network that have ports open which should not be.
- **Weak passwords:** Vulnerability scanners often include a password cracker which can check that users have set up strong passwords for their accounts.
- **Default accounts and passwords:** Some operating systems and applications have default accounts which should be disabled. For example, some SQL database software allows administrator accounts to be created with blank passwords. Scanners can check for this and alert the system administrator.
- **Sensitive data:** Some vulnerability scanners can monitor a network and look for sensitive data that is being transmitted without encryption, such as credit/debit card details.
- **Security and configuration errors:** Most vulnerability scanners will search for misconfigured software which may give opportunities for attackers. For example, due to user error or other reasons, a system may have anti-malware updates or the firewall switched off. The scanner will detect these types of error and alert the system manager.

Link

One of the best known vulnerability scanners is Nessus. You can find out more about it if you do an internet search for 'Nessus'.

Penetration testing

Vulnerability scanners check for issues that may allow an attacker to access a system. Penetration testing is a more active method of testing which involves actually trying to access the system using the same techniques that a hacker might use.

Packet sniffers

Packet sniffers are sometimes called protocol analysers and are used to capture and analyse network data packets. They can be used by both network managers and attackers to inspect and modify both the control information with a packet (the packet header) and the data.

Packet sniffers include filter tools that allow a system administrator to view certain types of packet and to check that they are not being manipulated in some way by an attacker. However, using a packet sniffer in this way would be quite a lengthy process

and a better solution might be to use a tool to automatically detect such packet manipulation, such as an intrusion detection system (IDS).

Link

One of the best known packet sniffers is Wireshark, which is available as a free download. You can find out more about it at www.wireshark.org

Security-based operating system distribution

Typically, in a large organisation with many desktop PCs to support, the IT department will create a complete operating system and applications distribution (such as Office and other software), including a full range of security applications features (for example anti-malware software) and test the distribution fully before rolling it out to all the desktop computers. They will often create a disc image so that the complete system can be easily installed on any PC without needing to install and set up all the individual applications and settings. This approach ensures that all systems are set up the same way and have the same security settings and user features.

Tip

Working in the IT support department of a large company will mean, as with many jobs, that you work within a team of people. This will mean working with people who have different skills and areas of expertise. Working well in a team requires a number of different skills. You need to play your part in the team and support the other members. You also need to communicate well, listen to and understand what others are saying and let them know what you are doing and any issues that you may have.

The effectiveness of the protection

Judging the effectiveness of the applied protection is not easy. IT security is not a subject where the 'wait and see' approach is likely to be effective. As new threats are developing all the time, a system administrator needs to be proactive in their approach to security. This means regularly checking that what they currently have set up is adequate and always looking for ways to improve it. The testing methods listed above provide some means of checking how effective protection measures are, but a system administrator needs to keep up to date with the latest threats and know how to defend against them.

The effectiveness of the protection can also be considered in terms of the impact it has on the usability and performance of the system. As mentioned earlier, it is important that the protection measures do not unduly restrict users and interfere with the day-to-day tasks that they need to complete as part of their jobs. One method of judging the effectiveness of applied protection methods is to view activity logs such as those that can be produced by firewall systems. The log provides evidence of when the firewall has blocked connections and, as such, shows the effectiveness of the rules it is applying.

Link

Setting up, viewing and interpreting the Windows® operating system Firewall log was discussed earlier in the section 'Firewall configuration'.

File access auditing can also be set to log file access failure events (i.e. when a user tries to access a file or folder that they do not have permission to access) which again shows that the protection applied to the file or folder is working. Anti-malware scan reports can also show when the software has dealt with malware infections and so demonstrate its effectiveness in protecting the system from them. The issue with using logs to check the effectiveness of the protection is that they only show where the protection has worked – if the protection is ineffective, then the logs will show nothing.

A system manager should always be looking for ways to improve the protection applied to the system. This should involve keeping up to date with new issues and vulnerabilities by using the many internet resources available on this topic. In this way, the system manager can stay informed about emerging threats and make recommendations for further improvements to the security protection. Systems managers may also consider the use of active testing techniques such as employing ethical hackers to attempt to gain access to the system and therefore help to assess the degree to which the system is protected.

Reflect

Assessment practice 7.2 has a number of stages to it, some of which are quite complex. When working on any big project, timescales are important and it will be essential that you complete the final assignment for this unit by the deadline date. It is important to develop time management skills.

Create an action plan for the assessment practice tasks, in which you break each task down as far as possible into subtasks and try to estimate how long each subtask will take. From this, you should be able to create a plan that sets completion deadlines for each task and shows you how much time you need to dedicate to each task. You will need to monitor your progress on the assessment practice activites against your plan. You may not be able to complete every task within the time allocated but it is important that you do not fall too far behind because it can be very difficult to catch up.

Assessment practice 7.2

C.P5 C.P6 D.P7 D.P8 C.M3 D.M4 CD.D2 CD.D3

Following the presentation you gave to the company directors about IT security, you have been asked to create a plan to protect a test system and then implement the plan. The system needs to be protected from both internal and external threats. You need to complete the following tasks.

- Produce a plan to protect the test system from IT security threats that explains how the protection methods will defend the system.
- Write a justification of the protection methods you have chosen in the plan, showing how they will defend the system from threats.
- Set up the test system and apply the security protection methods as described in your plan.
- Test the security protection methods you have applied and enhance them, based on the testing you have performed.
- Write an evaluation of the plan and its implementation that considers how effective the protection methods you have applied are likely to be.

Plan

- What is the task? What am I being asked to do?
- How confident do I feel in my own abilities to complete this task? Are there any areas I think I may struggle with?

Do

- I know what I am doing and what I want to achieve.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.

Review

- I can explain what the task was and how I approached the task.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

Further reading and resources

Websites

- BBC News Technology section – technology news, including cases of significant cyberattacks:
  www.bbc.co.uk/news/technology
- The Information Commissioner's Office website – gives case studies of action taken under the Data Protection Act:
  ico.org.uk/
- McAfee's security advice centre – provides useful and up-to-date information on security, virus attacks and viruses, written in a reasonably accessible manner:
  home.mcafee.com/advicecenter/
- Microsoft's security advice centre – features a regularly updated blog and FAQ on security issues:
  www.microsoft.com/security/
- Norton™'s security centre – provides articles on security, spam email, software piracy etc.:
  uk.norton.com/security-center/
- Bletchley park, the nation Museum of Computing and wartime code breaking centre:
  http://www.bletchleypark.org.uk/
- YouTube – search for video presentations on public-key encryption:
  www.youtube.com

THINK FUTURE

Imran Hussain, Apprentice IT technician

After school, I managed to obtain an apprenticeship in a medium-sized business working in the IT support department. Although I was aware that security is a big issue, I was quite surprised at the amount of helpdesk requests that are related to security. Security issues create a lot of headaches for users in all sorts of ways. We have to do a lot of password resets because users have forgotten their passwords, which is frustrating for both the users and us, but the company policy is that users must change passwords every 3 months. Some users feel like we are making life difficult for them, but the important thing for us is protecting sensitive data and the company systems.

After 6 months, I moved off first-line support which means no more password resets, but I now have to deal with much more complex and technical issues. One thing that I have learnt is that many security issues, such as firewall configuration and setting folder permissions, are very complex and unless you know what you are doing you can cause a lot of problems. I have learnt a lot, but there is still a lot more to learn. The management here are very concerned about IT security issues and regularly remind us that new, more sophisticated threats are likely to appear in the future so the situation is only going to get worse and we need to be constantly on our guard.

Focusing your skills

Planning to work in computing

Security is likely to be an issue in whatever computing role you have in mind for the future. If you are planning to work in technical computing roles such as programming or web development, or as an IT technician, then your understanding of IT security needs to go beyond the 'user' aspects of security, such as strong passwords and anti-malware measures. If you are working in web or software development security, then it is a particularly important issue because you need to understand how to build security into the products that you are developing.

- As IT security is such a dynamic area, you need to keep up to date with the latest security issues. Following technology blogs is one way of doing this. There are many different technology blogs – some of the best known are Techdirt, Guardian technology, Techworld and Krebs on Security.
- Do your own research into security issues and aim to develop an in-depth technical understanding of how some of the common threats, such as SQL injection, work. There is plenty of information on all the common threats available on the internet.
- If you are able to obtain it, work experience (or shadowing) has many benefits and will provide very useful experience that is difficult to obtain in any other way. It will help you to understand security issues from both the user's and the technician's perspective. As Imran has found in his work as an apprentice IT technician, users can often find security issues very frustrating, so you need to develop the interpersonal skills required to deal with users who may be upset and angry.

Getting ready for assessment

Amber is working towards a BTEC National in Computing. She was given an assignment which asked her to create a presentation about IT security threats and encryption for a company where she is working in IT support. This covers learning aims A and B. Amber shares her experiences below.

How I got started

First, I listed all the things I needed to do to complete the assignment and I created a time plan to ensure that I completed the assignment on time. I checked my progress against the time plan at regular intervals. Then I created an overall structure for the presentation listing the main things I needed to cover. These were:

- the different threats than can affect the IT systems of the company
- the principles of information security with examples relating to the company
- legal issues related to IT security and the company
- the principles of cryptography and how it can be used to protect data
- an analysis of the impact that security breaches could have on the company
- an evaluation of the methods the company can use to protect itself from IT security threats.

I collected all my class notes on these topics into a folder and divided them up into the sections listed above. I then chose a PowerPoint® slide template and created title slides for each section. I found that this approach worked well as it gave me a structure around which to build the presentation.

How I brought it all together

I worked through my notes for each section and used a highlighter pen to pick out the main points. I used these as the bullet points on each slide of my presentation, making sure that I didn't have more than about 5 or 6 small bullet points on each slide, otherwise the text would become too small to read. I also checked that I just had the main points on the slides and I rewrote the rest of the text from my notes into the slides notes section, rewording them as if I was talking to an audience. I also added some further screenshots to illustrate the text.

What I learned from the experience

I found that the notes I made in class were fine on some topics but too brief on others. I wish I had been more consistent in my note taking as I had to spend quite a lot of time doing research to find out about some topics which had been covered in class but for which my notes were not adequate. I found completing the last section – the evaluation – hard to write. My first attempt was really just an explanation of the techniques that the company could use to protect their systems. I had to take each protection technique and think hard about how effective it might be and what issues it might cause, such as performance problems or usability issues. In the end, I decided that this was hard to cover in the presentation slides themselves and that I needed to write the evaluation mostly in the slide notes.

Think about it

- Are you taking class notes and collecting the handouts that your teacher has given you? These will be really helpful when you come to write your assignments, so keep them safe and organised in a folder.
- There is a lot of information about IT security on the internet. You can use this in your assignments but check that it is up to date as this is a rapidly changing sector. You can use direct quotes only if you clearly reference them, otherwise you will need to rewrite the information you find in your own words.

Making a plan for completing your assignments is important. You must hand your assignment in on time, so creating a plan with timings can help you to ensure that you have everything ready in time.

Business Applications of Social Media 8

Getting to know your unit

Assessment
You will be assessed by a series of assignments set by your tutor.

Social media is a phenomenon of the internet age; nothing like it existed before. Its immense popularity has provided a new way for individuals, organisations and other groups to communicate with each other. For business users, social media provides a new way to interact with the public and customers. This unit is about the ways in which organisations can make use of social media, the methods and benefits of doing so and the issues and dangers that exist. Understanding how organisations can use social media successfully will be a useful skill to have when looking for work in the computing industry.

In this unit, you will explore different social media websites and consider the ways in which they can be used for business purposes, including the potential pitfalls. You will then develop and implement a social media plan for an organisation to achieve specific aims and objectives. You will also collect data on the organisation's use of social media and review the effectiveness of your social media plan.

How you will be assessed

This unit will be assessed by a series of internally assessed tasks set by your tutor. Throughout this unit, you will find assessment activities that will help you prepare for the live assessment.

For learning aim A, you will need to look into how social media can be used by organisations in general, while for learning aims B and C you will need to plan and implement the use of social media in a real or imaginary organisation. To pass this unit, you need to ensure that you have covered all the Pass criteria fully in the live assessment. To achieve the higher grades of Merit or Distinction, you need to ensure that you present evidence which meets the requirements of the individual assessment criteria. The Merit criteria require an assessment or a justification, so evidence which is merely a description will not be sufficient. For M2 for example, you will need to say why you made the decisions you did and why alternatives were rejected. Similarly, some of the Distinction criteria require an evaluation.

Assessment criteria

This table shows what you must do in order to achieve a **Pass**, **Merit** or **Distinction** grade, and where you can find activities to help you.

Pass	Merit	Distinction
Learning aim A Explore the impact of social media on the ways in which organisations promote their products and services		
A.P1 Explain the different ways in which an organisation can use social media to promote products or services to a target audience. **Assessment practice 8.1**	**A.M1** Assess the different ways in which an organisation can use social media to promote products or services to a target audience. **Assessment practice 8.1**	**A.D1** Evaluate the organisational use of social media to interact with customers and promote products or services to a target audience. **Assessment practice 8.1**
A.P2 Explain the audience profiles of different social media websites. **Assessment practice 8.1**		
Learning aim B Develop a plan to use social media in an organisation to meet its business requirements		
B.P3 Produce a plan to use social media in an organisation to meet its business requirements. **Assessment practice 8.2**	**B.M2** Justify planning decisions made, showing how the plan will fulfil its purpose and the organisation's requirements. **Assessment practice 8.2**	**BC.D2** Evaluate the plan and use of social media in an organisation against its business requirements. **Assessment practice 8.2**
B.P4 Review the plan with others in order to identify and inform improvements. **Assessment practice 8.2**		
Learning aim C Implement the use of social media in an organisation		
C.P5 Produce business-related content for an organisation using appropriate features of social media which meet the requirements of the plan. **Assessment practice 8.2**	**C.M3** Optimise the content, format and features of social media which meet the requirements of the plan. **Assessment practice 8.2**	**BC.D3** Demonstrate individual responsibility, creativity, and effective self-management in the planning and use of social media in an organisational context. **Assessment practice 8.2**
C.P6 Review data obtained on social media usage and interaction. **Assessment practice 8.2**		
C.P7 Assess the extent to which social media content and format improved search engine rankings **Assessment practice 8.2**		

Getting started

Social media has had a huge impact on our lives. Consider how social media is used by organisations. Do you ever click on adverts you see on social media? Do you follow/like any organisations? Have you joined any Facebook groups? Which ones did you join and why?

A Explore the impact of social media on the ways in which organisations promote their products and services

You probably already use social media sites, but this unit is not about the personal use of these sites. Instead, it is about the ways in which organisations can use social media to promote their products and services. However you have probably noticed that some things that you post on social media are more popular than others: they get more likes or shares, for example. Have you ever thought about why you like or share some posts but not others? For organisations, understanding what makes people interact with a post can really help them promote their organisation on social media effectively.

Social media websites

There are a wide variety of different social media websites, from the very well-known ones, such as Facebook and Twitter™, to the less well-known ones, many of which have a particular focus. Instagram, for example, is a social media website where users can upload images and videos taken with their phone, while LinkedIn® is used for making organisational contacts.

Promotion of an organisation using social media

Social media has become hugely popular and many millions of people have signed up to accounts on social media websites. Each site has its own particular features and the sites are constantly developing and adding new facilities to keep existing account holders engaged and to attract new members.

In general, each social media website has its own unique features and structure that make it different from the others.

Table 8.1: Key social media websites, their features, structure and target audience

Social media website	Key features and structure	Target audience
Facebook	• Connect to 'friends' – both real-world friends and acquaintances, and people you only know virtually through social media (often friends of friends). • Friends post status updates, photos/videos or links and interact with others by 'liking', commenting on posts or by sharing posts. • Dashboard view (when accessing the site on a PC) showing newsfeed, links to messages, events, friends, groups, pages, apps and adverts. • Groups – both public and private social groups for friends with a common interest, different communities. • Pages – clubs and societies as well as organisations may have a page which users can join to see their posts. • Newsfeed – a scrolling panel in the centre of the Facebook main display shows posts from you, your friends, groups and pages that you have liked. • Events – both public and private events can be organised with a dedicated page. • Targeted advertising and trending articles appear in a sidebar on the right of the page.	• Individuals who want to know what is happening in the lives of their friends and family. • People who want to follow particular interests using groups. • Organisations who want to promote their products and services.

Table 8.1: Key social media websites, their features, structure and target audience – *continued*

Social media website	Key features and structure	Target audience
Google+™	• Similar in concept to Facebook. • Users can organise friends into 'circles' which are groups of friends with which certain content can be shared. • Updates from a user's circles are displayed in the central 'stream' (similar to Facebook newsfeed). It is a two column display of friends' updates, and different circles can be selected using the menu bar. Communities allow users to join in a conversation about a particular topic of interest (similarly to Facebook groups). • Collections allow you to group your posts by topic. Other users can then follow a collection rather than everything you post. • Hangouts allow users to take part in multi-user video conferences.	• Individuals with a particular interest who want to follow a community of like-minded individuals.
Twitter™	• Post short messages (max 140 characters) known as Tweets which are seen by your followers. • Follow other Twitter™ users to see their Tweets. Twitter™ is often used to follow celebrities and keep up to date with what they are doing. • Retweet posts that are considered interesting to your followers. • Suggestions for accounts to follow and current trends on the left of the page.	• Individuals who want to know what is happening to celebrities and other people they are interested in.
Instagram	• Upload photos and short videos from your phone. • Use Geotagging to identify the location shown in the photo. • Filters can be applied to the photos to provide a variety of different looks. • Users can follow other people and see the photos they upload. • Users can connect their Instagram accounts to other social media accounts to share uploaded photos on those sites as well. • Simple newsfeed view, with a scrolling display of photos uploaded by the user or the accounts they follow. • Can only be accessed using mobile devices, not PCs. • Instagram has been owned by Facebook since 2012.	• Individuals who want to share photos with friends and family.
YouTube	• Upload videos, and create your own video channel which others can subscribe to. • Carry out simple editing on videos. • Subscribe to YouTube channels created by individuals or organisations with collections of videos on a particular subject. • There is a vast range of videos on YouTube, including music videos, and instructional videos covering almost every imaginable subject. • For an organisation with a physical product, YouTube is an excellent place for product demonstrations. • YouTube has been owned by Google™ since 2006.	• Individuals who want to upload videos. • Organisations that want to promote their product with a video. • People who want to watch music videos.
LinkedIn®	• Upload an organisation-related profile showing your professional experience and qualifications. • Groups related to professional interests. • Organisation pages providing a detailed profile of an organisation. • Personal users can link to other contacts and network. • A good place to look for job opportunities.	• Business people/ professionals and organisations that want to network and make contacts.
Pinterest	• Pinterest is a relatively new social media site which saw a big increase in users during 2014–15. • Allows users to create a scrapbook (or pin board) type collection of website links, images and videos. • Users can follow each other so they see their newly added content (so-called 'pins'). • Users can search and browse through any user's public pin boards.	• People who are interested in, for example, crafts who want to get ideas from others. • People who want to collect ideas on a particular theme.

Many social media sites work with each other, for example you can link your Instagram account to Facebook so that whenever you post a photo on Instagram, it automatically posts on your Facebook too. You can also post your YouTube videos on Facebook.

Discussion

What social media sites do you use the most? What features of those sites do you find most successful and engaging to use? What types of post get the most interaction (that is comments, likes and shares)? Discuss this with your class and create a list of the most popular features and try to identify why those features are popular.

Case study

Facebook

Facebook was founded by Mark Zuckerberg in 2004 and originally limited to university students, but since September 2006 it has been open to anyone over the age of 13. After 2009 Facebook grew rapidly, and it had 500 million active users in 2010 and by September 2014 it had reached 1 billion active members. Facebook is the second most visited website in the world after Google™ and the third most visited site in the UK (after google.com and Google™). After the US, the country with the largest number of Facebook users is India. However, Facebook is not as popular in every country as it is in the US and UK. In countries such as Japan, where local social media sites tend to be preferred, Facebook is still way ahead of other social media sites in terms of the number of users. In 2014, the percentage of online users who used Facebook was 71 per cent; its nearest rival was Linkedin™ with 28 per cent of internet users. (source: http://www.pewinternet.org/2015/01/09/social-media-update-2014/)

Facebook has had a huge impact on the way people communicate and relate to each other. It has changed the way that people stay in touch with individuals and advertise forthcoming events. Facebook has also had a political impact. For example, it is said that Facebook played a major role in the 2011 Egyptian revolution, allowing people to easily organise mass rallies.

In recent years, the growth of Facebook has slowed down, perhaps because it is reaching saturation in many countries, and other newer social media sites such as Instagram and Pinterest have grown more rapidly. However, these newer social media sites still have a long way to go to reach the number of active users that Facebook has.

How organisations can use social media

There are a number of ways in which organisations can use social media websites to support their business aims and needs and these are discussed in the following sections.

Promoting products and/or services and creating an image or brand

Organisations, just like individuals, can set up free profiles on social media websites. They use them to promote their products or services, and to describe their organisations to customers. Social media allows them the opportunity to develop an online community of followers. The aim is that the organisation will post interesting content that their followers (potential customers) will interact with by commenting, liking or sharing the content.

Many organisations use social media to help create a **brand** or image for their products. This is about associating the product in people's minds with the kind of image the organisation wants to portray. For example, an organisation might want to promote a healthy or sports-orientated lifestyle brand, or they want to associate the product with quality, success or luxury, or they might want to give a product a quirky, out of the ordinary, alternative image. The goal is to attract customers from their target market to think positively about the product or service, and ultimately purchase it.

Key term

Brand – an aspect of a product or service that distinguishes it from other similar products. It can include, among other things, a logo, colour scheme and name. Organisations often go to great lengths to develop a unique brand identity. Some of the best known brands in the world are McDonald's® golden arches, Apple®'s logo and Nike®'s 'swoosh'/tick.

Discussion

Think about some of the ways in which you could create a brand image for different products. For example, suppose you were asked to promote an expensive luxury car, a low-cost airline or personalised marshmallows. What type of image would you want to portray for these different products? How would you achieve that? Have a discussion about this with your class and decide on a brand image for a number of different products.

Communicating with customers

Using social media is an excellent way to communicate with customers, especially about unexpected or time dependant changes to the organisation's products or services. For example, many train and bus companies use Twitter™ to update passengers on delays, cancellations and route changes. Furthermore, the communication does not need to be one way as social media can be used to obtain customer feedback and to gather customers' opinions. This can be a much more cost-effective method of doing market research than traditional methods such as telephone or face-to-face interviews. It helps customers to feel that their opinions and ideas are valued by the organisation, as they feel that they have direct access to the organisation.

Customer service

Organisations can use social media to provide good customer service, by resolving queries and managing issues through their social media profiles. This is a method of communication that many people are familiar with and can access quickly and easily. When customers have questions or problems with a product or service they can quickly message or post on the Facebook page of the organisation or send them a tweet on Twitter™. For organisations to use social media successfully for customer service, they need to ensure that their organisation profiles are monitored regularly (or constantly) and that customers are replied to quickly, otherwise it may look as if the organisation does not care about their customers.

Features of social media websites tailored to organisational needs

There are a number of key features of social media websites that can be tailored to organisational needs. These are discussed in the following sections.

Social media profiles and advertising

Social media websites do not charge users (individuals or organisations) to create accounts. They make money by offering organisations paid advertising on their website. On Facebook, paid-for (sponsored) adverts appear on the right sidebar of the main newsfeed when using the full desktop version of the site. On Twitter™, paid-for tweets (promoted tweets) appear among the tweets from the people you follow in your Twitter feed. Social media websites offer advertisers special facilities such as adverts that only appear on the pages of users that meet certain **demographic** criteria, for example age or gender, or adverts that relate to the user's recent internet search history.

Key term

Demographics – measurements that are used to put people into different categories. One reason for doing this is to understand their likes and dislikes more easily.

Social media websites therefore need to attract organisational users and provide a number of features aimed specifically at them. Many sites provide facilities for organisations to create pages and profiles for the organisation (both Facebook and LinkedIn® provide this). The profile tells the public about the organisation – what they do, their business aims and where they are located. Some sites also provide tools which show detailed demographic profiles of the people who visit an organisation's pages and how they interact with them. The best known example of this is Facebook Insights, which is a powerful tool for gathering data on the effectiveness of your posts and the profile of your social media audience. Twitter™ and Google™ also provide their own analytics tools. Facebook Insights is covered in more detail later in this unit.

Link

For more on Facebook Insights and analytic tools for social media, see Data gathering and analysis.

If you want to place paid-for adverts on Facebook, you can choose the audience you want the adverts to reach, in terms of demographics. For example, if you are a plumber, you really only want to advertise to local people so it would be useful to be able to target people based on their location. Figure 8.1 shows the Facebook advert set-up page, where you make these selections.

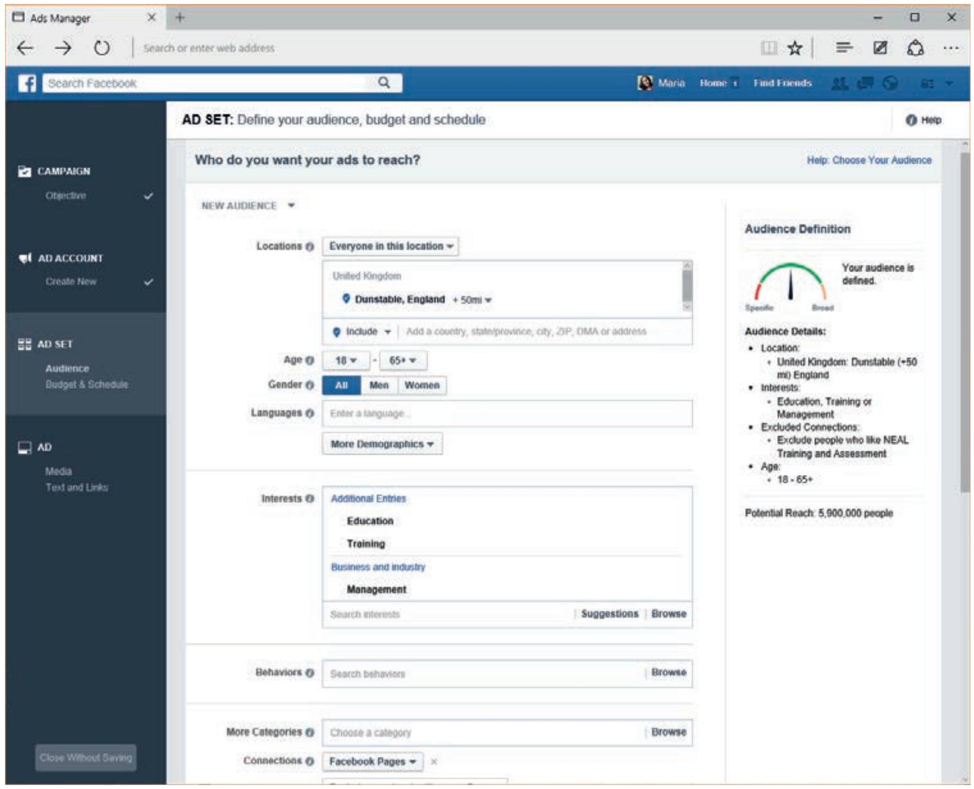


Figure 8.1: Facebook advert targeting set-up

You may have noticed that if you search for a product on a website like Amazon, adverts for the product you have searched for start appearing on your Facebook page. Providing a link to a user's previous e-commerce site search history, and displaying related adverts on their page, is an organisational feature that Facebook offers to advertisers. The aim is to remind users of items they have looked at, but not yet purchased, in the hope that they will now purchase them. However, paid-for advertising is only a small part of what an organisation can use social media for.

Website and mobile device integration

Many people access social media via mobile devices rather than from laptops or PCs and this is a growing trend as people like to be able to communicate and stay 'plugged in' on the move. Most social media websites have mobile versions of their sites which are optimised for small displays, touch screen interaction and low processing power. Mobile integration is important for organisations because it allows users to locate local services and may help organisations pick up passing trade. For example, restaurants and fast food outlets have the opportunity to attract trade from visitors to the area. Some specialist websites, such as Foursquare™, specialise in providing location-related information.

Link

For more on mobile device integration, see 'Mobile device integration' in *Unit 17: Mobile Apps Development*.

Search engine optimisation

Social media have an important relationship with search engine optimisation (SEO). SEO is the name given to a range of techniques applied to a website to attempt to improve the ranking of the website (how close to the top it appears) in unpaid search results (in other words, the standard Google™ or Bing search results rather than paid-for adverts). The theory is that when customers search for a type of product or service, they will be more likely to contact the one that appears first in the search results and are very unlikely to contact organisations that do not appear on the first page of results.

All organisations want to improve their search engine rankings. There are lots of things an organisation can do to their website to achieve an improvement in search engine ranking, but they should also have and maintain a Google+™ (Google Plus) account. As Google+™ is owned by Google™, the most widely used internet search engine, using a Google+™ account is likely to help improve an organisation's search engine ranking.

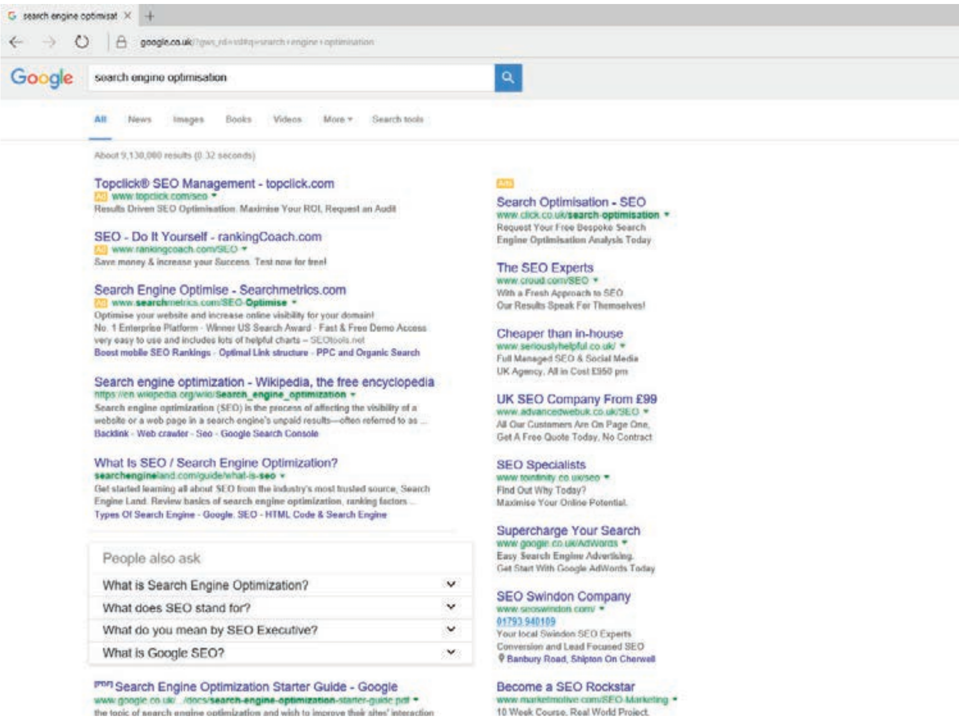


Figure 8.2: The results of a search for 'Search engine optimisation' (Google and the Google logo are registered trademarks of Google Inc., used with permission)'.

Some of the most important things an organisation can do to improve the search engine ranking of their website are listed below.

- Decide upon and use appropriate **keywords** within the content of the website.
- Update web pages regularly. Search engines assume that websites which are not updated regularly are less likely to be relevant to people searching for something.
- Use the most relevant keywords in the URLs and web page titles and to describe the site.
- Ensure that website text is properly spelt and uses correct grammar. Search engines assume that websites which have many spelling errors or bad grammar are not likely to be relevant.
- Encourage other people to add links from their website to yours (sometimes called 'inbound' links). If other organisations think your website is worth linking to then search engines assume that this is because there is a lot of useful information on your website and that it is worth others being directed to it.

Key term

Keywords – words that identify the key things that an organisation has to offer to customers and are likely to be used by potential customers when carrying out an internet search for a product or service.

Link

Keywords are covered in the section entitled 'Keywords', in 'Content planning and publishing'.

SEO is a huge topic, and large organisations will spend a great deal of time and effort ensuring that their website is fully optimised. Search engines themselves provide

guidance on how to achieve good results with their products. For example, search for 'Google webmaster guidelines' to see the advice that Google™ offers to websites on how to optimise their content.

Audience profile

An important concept in understanding how organisations can use social media is that of the 'audience profile'. The audience profile of a social media website describes the nature of the people who have registered profiles with the site, in terms of their age group, gender, geographical location and so on. Different social media websites have different audience profiles and, over time, the profile of a site's users can change.

Research

Run an internet search for 'social media audience profile'. This should produce some interesting results, but make sure that you are looking at recent information and also be aware that many results may only show data for the US. How do the profiles of sites differ? Pick a variety of different audience profiles (age, gender, location etc.) and find the site which matches the profile the closest.

For example, in 2015 research suggested that more men than women used Twitter™ but the reverse was true for Instagram, which had more female users. Instagram also has a younger audience profile than most other social media websites. Pinterest has a much higher proportion of female users than any other mainstream social media website. Another example of the way in which social media usage has changed over time can be seen from research carried out in the US into the use of social media by teenagers. During 2015, the number of teenagers using Twitter™ fell, while the numbers using Instagram and Snapchat™ rose, although Twitter™ remained the second most important social media network for US teenagers (behind Instagram).

LinkedIn®, on the other hand, with its focus on professional business relationships, was more popular with the over 50 age group than any of the other popular social media websites.

A key benefit to an organisation of using social media is the ability to identify the audience profile that they have attracted. Facebook, Twitter™ and other social media websites provide tools which show the profile (in terms of demographics) of a page's audience. This is usage data which indicates the profile of followers. These analytic tools, such as Facebook Insights, also show organisations the effectiveness of individual posts in terms of the amount of interaction generated (that is, likes, comments or shares). This type of data is invaluable for marketing, but is either not available or difficult and expensive to collect using traditional types of media (TV, radio, newspaper and magazine adverts) and market research. Hence, social media has fast become a powerful tool for organisations in terms of promoting their products and services, because it allows them to easily understand who their target audience is.

Organisational uses of social media for business purposes

There are so many people using social media on a daily basis (there are more than 1 billion active Facebook users) that organisations have potential access to a huge market for their products and services. Any organisation can create a profile on any number of different social media websites but a profile on its own is not enough – they need to encourage people to follow their pages and must interact with their potential customers. In other words, their profiles cannot be idle. They should be updated with regular new posts and they need to be engaging to invite customer interaction.

Content formats

Social media websites allow organisations to post a variety of different content. You can post text and images on most social media websites, and many sites such as Facebook and YouTube also allow you to post videos. Facebook works well for short promotional videos. However, YouTube is a better choice for instructional videos because they stay in one place in their YouTube channel, whereas videos on FaceBook move down your timeline. The different social media websites can complement each other. For example, if an organisation provided customer service over Facebook and Twitter™ and a customer messaged asking how to do something, the reply could link them to an appropriate video on the organisation's YouTube channel.

Almost all social media websites allow you to include web page links in posts. This allows organisations to link their social media with their own website. For example, many organisations that use Twitter™ tweet a brief comment (remember tweets are limited to 140 characters) alongside a link to a web page or blog page article.

The Facebook Poll app allows posts to include a poll. This can be used in a number of ways, for example it can collect customer feedback on a new product or version of a product. There is also a Facebook Quiz app which can be

used to create a simple knowledge quiz or a 'personality quiz'. For example, a 'personality quiz' could be used by a jeweller, who creates a 'What kind of jewellery suits your personality?' quiz that suggests different types of jewellery to potential customers based on the user's answers to several questions.

Content focus and developing an audience

As well as being able to post different formats of content, organisations can use social media to post content that focuses on a wide variety of different purposes.

- To be informative – that is, to describe an organisation's history, aims and objectives in order to help potential customers buy into an image/brand. Social media posts can also inform customers of important information quickly (e.g. changes to train arrival/departure times).
- To promote – that is, to advertise and promote the products, services and image/brand/lifestyle ethos of an organisation.
- To entertain – depending on the organisation, posting humorous or other entertaining and engaging content could appeal to potential customers.
- To make offers – organisations use social media to make special offers to customers. For example, they might offer 24 hour discounts to customers who have liked and shared the organisation's page or offer a prize to the ten thousandth person who likes their page.
- To provide customer service – this will normally be in response to customer posts on their page. Responding quickly and directly to customers on social media and solving their problems will make an organisation look efficient and effective.

Ultimately, the main reason for organisations to use social media is to attract more customers and to make more money by selling more products or services. However, an organisation that constantly uses a 'hard sell' **direct advertising** approach on social media may end up annoying its followers rather than encouraging them to purchase. A much better (and ultimately more profitable) approach to developing an audience for your organisation is by mixing posts that promote products and services directly (and special offer posts) with informative or entertaining content, related to the area of business or sphere of life that the organisation operates in. This approach is a form of **indirect advertising**. For example, if a pet grooming business had a Facebook page, they could include posts related to issues to do with pets (such as pet health, grooming and food), photos of unusual pets, or links to amusing pet videos and other similarly pet-focused content that would be of interest to their target audience of pet lovers.

Key terms

Direct advertising – where an organisation uses adverts which tell you to buy their product or sign up to their service. They use a direct approach by simply telling you what the product/service is and suggesting that you purchase it.

Indirect advertising – is more subtle than direct advertising. It attempts to create a positive attitude towards the product/service in the mind of the customer through sponsorship (e.g. an organisation sponsors a particular TV programme), product placement and other methods (such as those that social media uses) to try to create a relationship with the customer.

This type of engaging content, focused on issues which will be of interest to their target audience, will help promote their social media streams and encourage people to like or share their posts (on Facebook) or retweet them (on Twitter™). It will also help to create a positive image of the organisation in the minds of the people who follow them, giving the impression of an organisation that is not just interested in selling, but has a genuine interest in the subject area and is knowledgeable about it.

Keywords

The choice of keywords is an important concept that relates both to an organisation's website and the content they post on social media. In most cases, when someone is looking for something online, they use a search engine such as Google™ or Bing. It is very important for an organisation's success on the web that, when someone searches for the product or service they provide, their website or social media pages are listed at or near the top of the search results. Keywords relate to the likely search strings (the text a person searching for the product or service types into the Google™ or Bing search box) that potential customers would use. It is important to make sure that these keywords appear in the social media posts and profiles that an organisation creates and are used in the URLs and page titles of their website, as it will increase the likelihood that the organisation's website and social media profiles will show high up in search results. For example, the keywords for a freelance editor's website and social media are the services they offer, such as 'proofreading', 'copy-editing' and 'project management', terms associated with their work such as 'publishing' and 'editor', and the name of the business.

Developing contacts

It is also important to develop contacts via social media. Organisations can use social media to keep up to date with their industry and with suppliers and competitors. By following their suppliers, competitors and industry experts on social media websites like Twitter™, an organisation can stay informed and also use this information to plan their marketing, product development and keep up to date with developments in technology. This can be a much better method of keeping up to date that having to trawl through a range of magazines and websites. An organisation can further develop their contacts (their online community) by sharing or retweeting relevant posts from other organisations or individuals. This will hopefully encourage other organisations to return the favour and occasionally share or retweet a post from them. Remember that one of the aims of a organisation's social media page is to provide material that will be of interest to their target audience, so other organisations' and industry experts' blogs can provide a useful source of the type of content that is needed.

Research

What kind of organisation would you like to run? Who would be the target audience? What kinds of content would they be interested in? Search on Facebook for groups related to that area of interest. Search Twitter™ to see if you can find any relevant accounts of related businesses, organisations or industry experts. Search the internet to see if there are any blogs related to these interests. All the content you find could form the basis of potential social media posts for your organisation.

Social media and organisation websites

Most organisations will have a primary website to promote themselves, as well as social media pages. Ideally, the two should link closely with each other. The colour scheme, text (including profile information) and graphics should be consistent across both the organisation's website and all of their social media pages. Social media is dynamic, with content posted today and forgotten tomorrow, but a website is more stable. Although it should be updated regularly, this is the place where more static, detailed content should be presented. The organisation's website should include buttons to link to its various social media pages. Many social media websites provide HTML code to create these buttons and make them work, which organisations can use on their websites. Figure 8.3 shows Twitter™'s website resources page from which you can copy the HTML code required to put various types of Twitter™ button on your website.

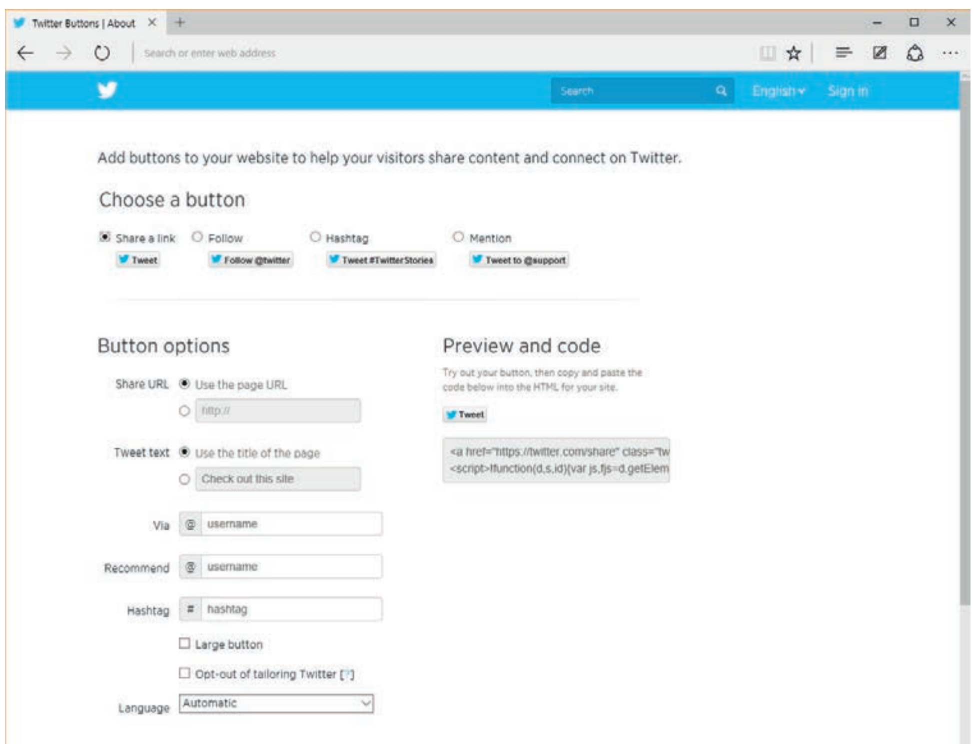


Figure 8.3: Twitter™ enables organisations to add Twitter™ buttons to their website that link to their Twitter™ account

Where relevant, an organisation's social media posts should contain links to the their website. For example, an organisation might post a 'teaser' photo of a new product on their social media pages, with a 'find out more' link to a page on the organisation's website, where more detailed product and service information is provided and can be purchased.

Another way to link an organisation's website to its social media pages is to embed social media feeds into the website itself. Facebook and Twitter™ provide HTML code that can be used to embed a newsfeed into a sidebar or another area of a website.

Risks and issues

While social media provides many excellent opportunities for organisations to promote their products or services, using social media is not without its pitfalls.

Time constraints and return on investment

Creating and running the social media presence of an organisation is time consuming, as it takes time to build up interest in an organisation on social media and results will not be achieved overnight. An organisation would probably need to have at least one person who dedicates a considerable amount of their time to social media marketing, and it may be that a small or even medium-sized organisation could not justify this expense when a positive effect on sales is not guaranteed.
When any organisation invests time or money into a project, they need to consider the return on investment. In other words, they need to weigh the time and money spent on their social media marketing campaign/customer service against the potential increase in sales that they might be able to achieve. It is quite hard to make this judgement prior to setting up a social media profile, so an organisation might need to run a trial in which they try out the use of social media over a reasonable period (say 6 to 12 months) and then review the results to see what the benefits to the organisation have been (that is, whether there has there been an increase in sales or their customer service ratings have improved).

Negative comments

An issue faced by all organisations using social media is that people may make negative comments on their social media pages. Every organisation has customers that make complaints, but, when customers complain via social media, other customers can see the complaints and this can be potentially very damaging for the organisation's reputation. The only effective way to deal with negative comments is for the organisation to respond to them promptly and correctly. Often the correct way to deal with a negative comment is to offer an apology and some kind of solution to the problem. Dealing with negative comments promptly requires someone in the organisation being able to monitor all the organisation's social media pages regularly and deal with any comments, preferably within hours of them being made. This significantly adds to the cost of using social media for the organisation, and the question of return on investment is raised again. However, organisations that deal with customer service issues promptly and effectively are likely to gain a good reputation.

Unforeseen consequences

There can be unforeseen consequences of posting content to social media. There is a very wide range of different opinions and beliefs among the general public, both within the UK and across the world. Sometimes, content which you find engaging and appropriate is felt to be inappropriate by others. It is very easy to post something on social media which you intended to be witty or sarcastic, but is viewed by others to be insensitive, crass or insulting. This unforeseen consequence of a post on social media can be very damaging to an organisation, particularly if the post and the offence people have taken to it goes **viral**.

There is also a danger that members of staff who post content on an organisation's social media pages might make comments or express opinions that are damaging to the organisation. This may be unintentional, and might be done with the intention of being friendly and helpful to a customer. For example, an employee dealing with complaints about a particular product, might, in an attempt to calm an angry customer, agree with the customer that the product is not ready for sale or contains serious flaws. As this will be seen publicly on social media, it could potentially be damaging to sales of the product and the organisation's reputation.

Key term

Viral – a social media post is said to have gone 'viral' if it is shared by numerous people who themselves share it (and so on). As more and more people share the post, the number of people who see it can increase into millions. Having a social media post go viral is often considered a good thing unless, of course, it is regarding a negative issue about an organisation.

Case study

Urban Outfitters

In October 2012, a serious storm called Hurricane Sandy hit the east coast of the US. It left 71 people dead and caused 71 billion dollars' worth of damage. US clothing retailer, Urban Outfitters, tweeted: 'This storm blows (but free shipping doesn't)! Today only...' with the hashtag #ALLSOGGY. Although at the time of the tweet the full extent of the devastation caused by the storm was not yet known, it was a tasteless and inappropriate tweet as no one wants to see an organisation trying to make money out of peoples' misfortune. This is widely quoted as an example of a social media public relations disaster.

Increased vulnerability to cyber criminals

Using social media can increase the number of security risks for an organisation. Cyber criminals are always looking for ways to attack and defraud organisations and social media provides an opportunity for them. These risks can take a number of different forms.

- **Malware** – Cyber criminals may try to trick social media users into installing some kind of malware. For example, by posting something interesting and inviting, they encourage users to click on a link which then downloads malware. Alternatively, cyber criminals may trick users into giving away their login credentials. Organisations must therefore make sure that their employees are aware of these dangers and take extra care when using social media and engaging with customers. The organisation must, of course, also ensure that their anti-virus software is always up to date and other security measures are in place.
- Ransom/blackmail – An organisation, by raising its profile on social media, may attract the attention of cyber criminals who then target the organisation. This could potentially lead to a variety of attacks including those designed to extract a ransom from the organisation, for example a **denial-of-service (DoS) attack** on the organisation website (which is essentially blackmail).
- Organisation-sensitive and personal information – Members of staff may inadvertently give away organisation-sensitive information or personal information (about members of staff or customers). Alternatively, the information is stolen by cyber criminals who trick staff into giving away this information by messaging them through the organisation's social media pages with seemingly innocent requests. To protect themselves against these, and other, risks associated with social media, organisations need to ensure that their staff are fully trained in the use of social media and are aware of what they can and cannot do and say. This is usually spelt out in a social media policy document, which will be covered in detail later in this unit.

Link

To learn more about social media policies, see Developing a social media policy.

Key terms

Malware – an umbrella term for a range of different types of software that have a malicious intent. Malware includes viruses, Trojans and spyware, among others.

Denial-of-service (DoS) attack – an attack on an organisation's website which involves sending so many bogus requests to the server where the website is hosted that it is overwhelmed and cannot respond to legitimate requests. The likely purpose of the attack is either revenge or blackmail.

Reflect

As you work on through this unit, it is important to demonstrate the kind of behaviour that has a positive impact on others and on your learning (that is, to be professional). Professionalism mean things like showing a clear division between your personal and professional use of social media, and completing work fully and on time. You may also have to work with others in your class, perhaps providing feedback for them on their social media posts and by playing the role of a customer commenting on posts they have made. You need to do this in a supportive and professional manner, demonstrate good etiquette and be polite and reasonable at all times.

PAUSE POINT Make a list of the risks and issues that an organisation of your choice would face when using social media.

Hint Look back through the topics covered in this section and think about how they relate to an organisation of your choice.

Extend For each risk or issue that you identified, work out how your chosen organisation would mitigate the risk or deal with the issue.

Assessment practice 8.1

A.P1 A.P2 A.M1 A.D1

You have been asked to give a presentation to local small-business people about how they might be able to use social media to promote and benefit their organisations.

You need to prepare a presentation including speaker's notes for the business people, in which you:

- explain the different social media websites and the audience profiles they attract
- provide an assessment of the different ways in which an organisation can use social media to attract a particular target audience
- evaluate the organisational use of social media to interact with customers and promote products or services to a target audience.

Plan

- What is the task? What am I being asked to do?
- How confident do I feel in my own abilities to complete this task? Are there any areas I think I may struggle with?

Do

- I know what it is I am doing and what I want to achieve.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.

Review

- I can explain what the task was and how I approached the task.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

B Develop a plan to use social media in a business to meet requirements

Using social media offers organisations many benefits but, as outlined already, there are also risks and issues. Therefore it would be unwise for an organisation to start using social media without first forming a plan. A carefully managed social media campaign can avoid many of the pitfalls and can also help the organisation to clearly identify if the effort, time and money of running the campaign will be worth it.

Social media planning process

For a small business, such as a sole trader, the burden of planning and running a social media campaign will fall to the business owner. As part of the planning process, they should think carefully if they are going to have time to do all the things that will be required. In a medium or large organisation, there may well be one or more people dedicated to running the social media campaign, and, in a really large organisation, there could well be a whole team. Whether a whole team is involved in the planning or just one person, there are a number of things which need to be decided and planned.

- The specific requirements of the organisation – What are the organisation's requirements? What is the organisation intending to use social media for and what is it hoping to achieve by doing so? In a larger organisation, these questions will probably need to discussed with the sales and marketing team(s).
- Content planning and publishing – What kind of content will the organisation post? How will the organisation obtain content to post, such as images and videos? What social media websites will the organisation have pages on? When and how frequently will the organisation post content?
- Developing online communities – How will the organisation go about developing an online community of followers? How will the organisation keep followers engaged once it has them?
- Enforcing social media policies – What will the organisation include in its social media policy? How will it ensure that employees comply with the policy?

Each of these areas of planning will now be looked at in more detail.

Organisational requirements

There are many ways in which an organisation can promote itself, for example newspaper, TV and radio, to name but a few. Before an organisation embarks on a social media campaign, it should consider what the benefits might be of using social media to promote its products and/or services compared with using other traditional methods.

Using social media may not suit all types of organisation. For example, a geographically-limited organisation selling a low-value product which cannot be easily differentiated from other similar organisations except on price (such as a small grocery store) might not get much benefit from using social media. However, an organisation that sells a product which can be shipped worldwide and which can be differentiated from other similar products (perhaps by uniqueness or quality), such as a jewellery maker, is more likely to benefit from using social media.

Requirements for the use of social media

Organisations need to consider the purpose of their use of social media, that is, what does the organisation want to get out of it? For example, do they want to use social media for the direct selling of a product and service, to create an image/brand, for customer service, to drive traffic to the organisation website or a combination of all of these? Without a clear idea of what the organisational requirements are, the social media campaign will not have a clear focus and is therefore not likely to be successful. Initially, it is better to have one or two simple and clearly defined requirements, rather than trying to do a wide range of things. Once the main requirements have been decided, the organisation also needs to consider who its target audience is, which social media websites are best suited to the organisation's products or services and what the business aims of the organisation are.

Selection criteria – matching social media websites to organisations

Even if an organisation is suited to social media, not every social media website will suit every organisation. With so many different social media websites, organisations, especially small ones with limited resources, need to select the right social media websites for their business needs. They should focus their efforts (time and money) on those social media websites whose site profiles best match their aims.

- Facebook's dominant position in the social media market means that, for many organisations, a Facebook presence is a must.
- However, an organisation offering professional services to other organisations would probably want to include LinkedIn® in their social media pages. For individuals such as freelancers who offer professional services to other organisations, a LinkedIn profile will be of more use than a Facebook page.
- If an organisation's products or services target a largely female audience, then they might want to consider using Instagram and Pinterest as these social media websites have a higher proportion of female users than most of the others.
- An organisation that has a product or service which is highly visual in nature (e.g. a jeweller, fashion designer or wedding photographer) would be wise to look at social media websites such as Instagram for promotion because it focuses on images.
- YouTube would be a good choice for an organisation with a physical product which requires demonstration because it is a good place to showcase videos. For this reason, setting up a YouTube channel would also benefit video production and animation organisations.

Any organisation should monitor the performance of its social media pages to ensure that its efforts are not wasted. If some pages constantly outperform others, then efforts should be focused on the most effective social media pages.

Success criteria and targets

It is important that the organisation considers how it will measure the success of using social media and the business targets that it will set. Measuring success and setting targets is important because the organisation needs to be able to prove that the time and money spent by the organisation on using social media is having measurable benefits. Identifying criteria for measuring success means that an organisation needs to understand what the measurable business benefits would be for that organisation of using social media. Largely, this comes down to fulfilling the requirements for using social media that the organisation decided upon. The extent to which they are successful in fulfilling those requirements is most easily measured by setting targets.

It is unlikely that any organisation will see significant results in a short timescale. It can take some time to build up followers and to develop an online community, particularly for new organisations or ones that have not used social media before. So the initial success criteria and targets that an organisation sets itself should probably be quite modest: for example, aiming to achieve a specified number of 'likes' or followers within 6 months of starting the campaign. If an organisation is using multiple social media websites it should set a target for each so that it can measure the performance of each social media stream individually. The target set should match the site's own terminology for interaction. For example, on Facebook, 'likes' are normally considered to be a good basic measure of popularity, so a good example of a target for using this site might be 'achieve at least 500 page likes over the next 6 months'. Twitter™'s popularity, on the other hand, is often measured by the number of followers an account

has. However 'likes' or the number of followers does not really measure interaction by users on individual posts, so setting targets for interaction such as (for Facebook) 'achieve an average of 20 likes, comments or shares per post' can be used. Ideally, the organisation 's social media targets should be 'SMART' (Specific, Measurable, Achievable, Realistic and Time-bound), so targets such as 'improve social media interaction' are not SMART, but the target 'increase the number of Twitter™ followers from 50 to 100 in 3 months' is SMART.

Timescales and responsibilities

When planning a social media campaign, it is important that the organisation decides who is going to do what within the team (if there is one) and creates an outline action plan for completing the tasks (that is, defines who is responsible for each task and establishes the timescales involved). There should be someone within the team who is responsible for setting the overall strategy for the campaign. They will probably work closely with the marketing and sales teams. There also need to be people who will develop the content to implement the strategy (that is, set up the profiles/pages and produce content that will be posted) and there should also be people who monitor the social media pages and respond to comments.

If an organisation decides that they do not have the expertise in-house to develop a social media marketing campaign, they might bring in an external social media consultant to help them through the process, although this would obviously add to the cost.

The end result of the planning process is likely to be a proposal document outlining all the things listed above, along with the action plan. In a small organisation or for a sole trader, this might just consist of some simple notes, but, in a large organisation, the proposal might need to be a much more formal document which is signed off (approved) by higher management who would need to agree the costs and resources involved.

	A	B	C	D	E	F
1	Day number	Day of week	Main task	Secondary task 1	Secondary task 2	Details/notes
2	1	Monday	Write and post blog post 1	Share blog post on social media		
3	2	Tuesday	Brainstorm ideas for future blog posts			
4	3	Wednesday	Research possible guest authors	Get in touch with one or two possible guest authors		
5	4	Thursday	Write and post blog post 2	Share blog post on social media		
6	5	Friday	Start outreach campaign to connect with other bloggers			
7	6	Saturday				
8	7	Sunday				
9	8	Monday	Write and post blog post 3	Share blog post on social media		
10	9	Tuesday	Confirm first 2 guest authors if possible	Get in touch with other possible guest authors if necessary	Set up schedule for guest authors	
11	10	Wednesday	Research possible guest authors			
12	11	Thursday	Write and post blog post 4	Share blog post on socia media		
13	12	Friday	Share historic post on social media			
14	13	Saturday				
15	14	Sunday				
16	15	Monday	Write and post blog post 5	Share blog post on social media		
17	16	Tuesday	Continue outreach campaign to connect with other bloggers			
18	17	Wednesday	Share historic post on social media			

Figure 8.4: An action plan

Content planning and publishing

One of the biggest challenges in creating an effective social media campaign is to create content which not only reflects the organisation's image and products/services, but is also engaging for the audience. Users engage with content that they find interesting, useful or amusing and, by being engaged, they are more likely to interact with the content in some way: that is, 'like' it, retweet it, comment on it or share it.

Target audience

When planning a social media campaign, one of the first things you need to do is to ensure that you have correctly identified your target audience. That is, you need to understand the types of people who are interested in your products or services in terms of demographics (for example, age, gender, interests or income). Once you have done this, you can think about how you will engage your target audience through your social media content.

Link

For more on target audience, look back at 'Audience profile'.

Simply posting special offers or details of products and prices is not really enough to attract an audience, and using social media for direct advertising is not likely to be very effective. A much better approach is to use indirect advertising and to consider what kinds of things are likely to engage your target audience. Indirect advertising via social media involves posting content that is informative and interesting to the target audience, rather than just sales-orientated. The organisation needs to consider what kinds of content the people who are likely to buy their products/services (that is, their target audience) might be interested in. For example, a small business that sells audio equipment (e.g. speakers and amplifiers) might post links to news articles about new advances in audio technology, the resurgent interest in vinyl records and people who have unusual or super-expensive audio set-ups. This content is likely to engage their target audience, who will then be more likely to interact with the content and ultimately to consider purchasing products from this audio equipment business.

Some of the content that an organisation will post will be in direct response to the comments made by others on their pages or related to current events. It is obviously difficult to plan this content beforehand. Nonetheless, other content will need to be planned, even if only in outline, so that the plan can be adjusted if things change.

Some organisations, if they can afford it, will enlist the help of advertising or social media consultants to help them develop content.

Link

For more on focusing content to match a target audience, see 'Content focus and developing an audience'.

Keywords

The need to include keywords in profiles and posted content has already been mentioned, but how does an organisation choose the most appropriate keywords? The question an organisation needs to consider is 'what will people type into the search box when they are searching for the product or service we provide?'. This might sound like a simple question, but it is often more complex that it might at first appear and there might be a range of different search strings used. For example, imagine a small business providing wedding photography and video services in the Cambridge area. What would people type when searching for that kind of service? Here are some possibilities:

- wedding photographer in Cambridge
- wedding photos in Cambridgeshire
- video and photos for weddings
- photographer for weddings
- wedding videographer
- wedding photography.

Therefore it would be important to ensure that the small business used the following keywords (which all relate to their core services, industry and location) prominently on their website (that is, in their page titles and URLs) and in their social media profiles. The keywords would be: wedding(s), photographer, photography, video, videographer, Cambridgeshire.

Google Adwords™

One very useful tool for identifying the most likely search strings to be used is Google Adwords™. The service is aimed at people who want to use paid adverts on Google™, but you can research search strings for free. Researching keywords is useful because it helps you to understand the most popular search strings that people use to search for a particular organisation's products or services. Once you know the most popular search strings, you can ensure that you use the correct keywords in your social media profiles and on your website.

Research

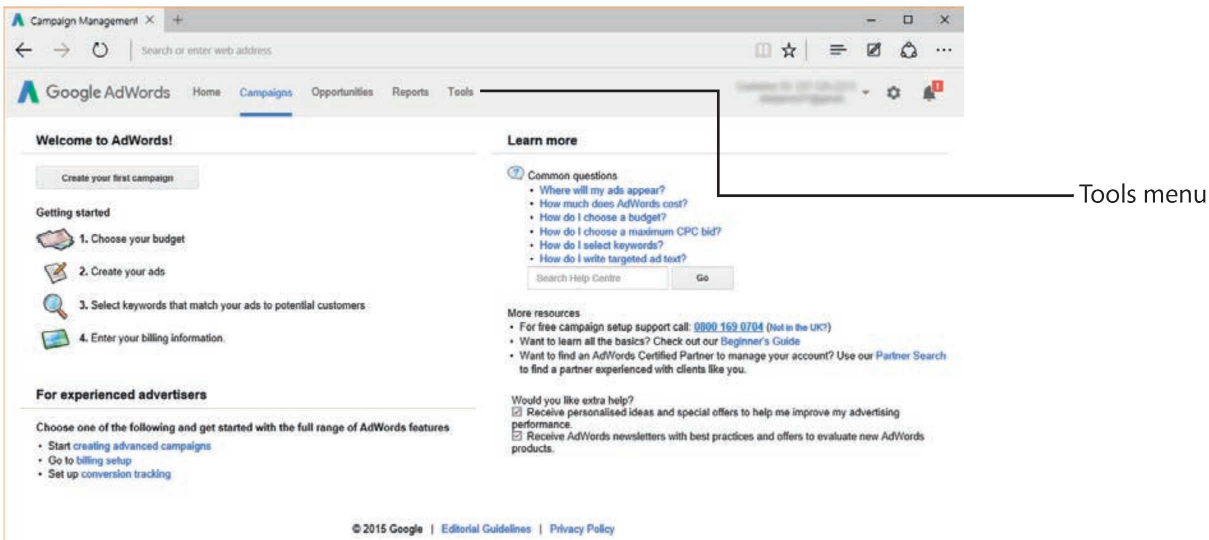
You may have considered the idea of running your own organisation. How do you imagine people might search for your organisation on Google™? Use Google AdWords™ to research the search strings that people actually use. Based on your research, choose four or five keywords that you would use for your organisation.

Step by step – Google Adwords™

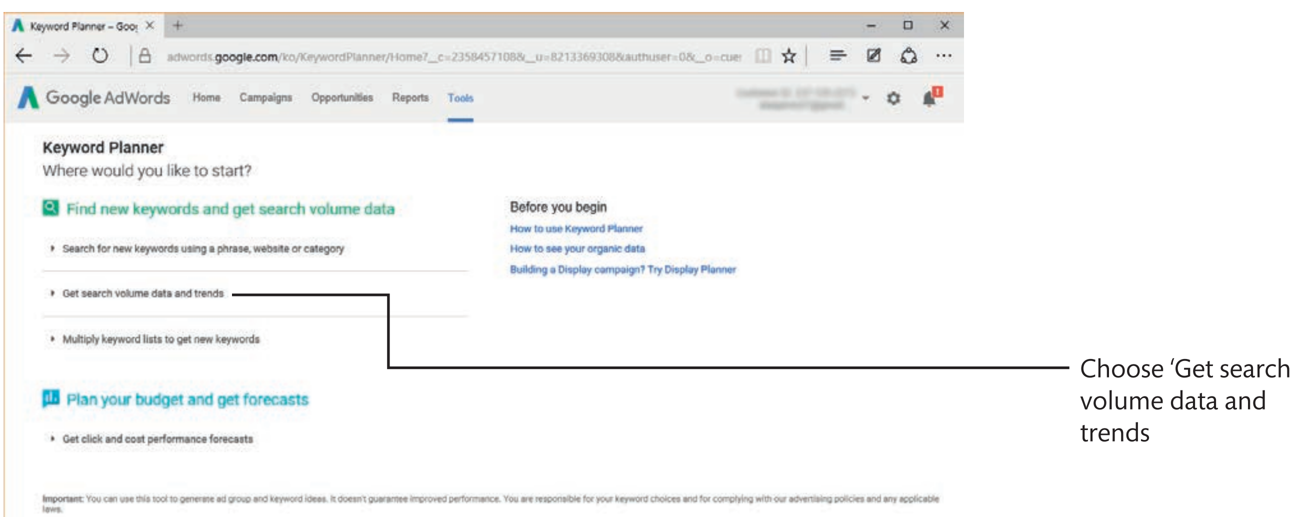
6 Steps

1 You need to register with Google Adwords™ (if you already have a Google™ account, you can just use those login credentials), then you can use the Keyword planner tool.

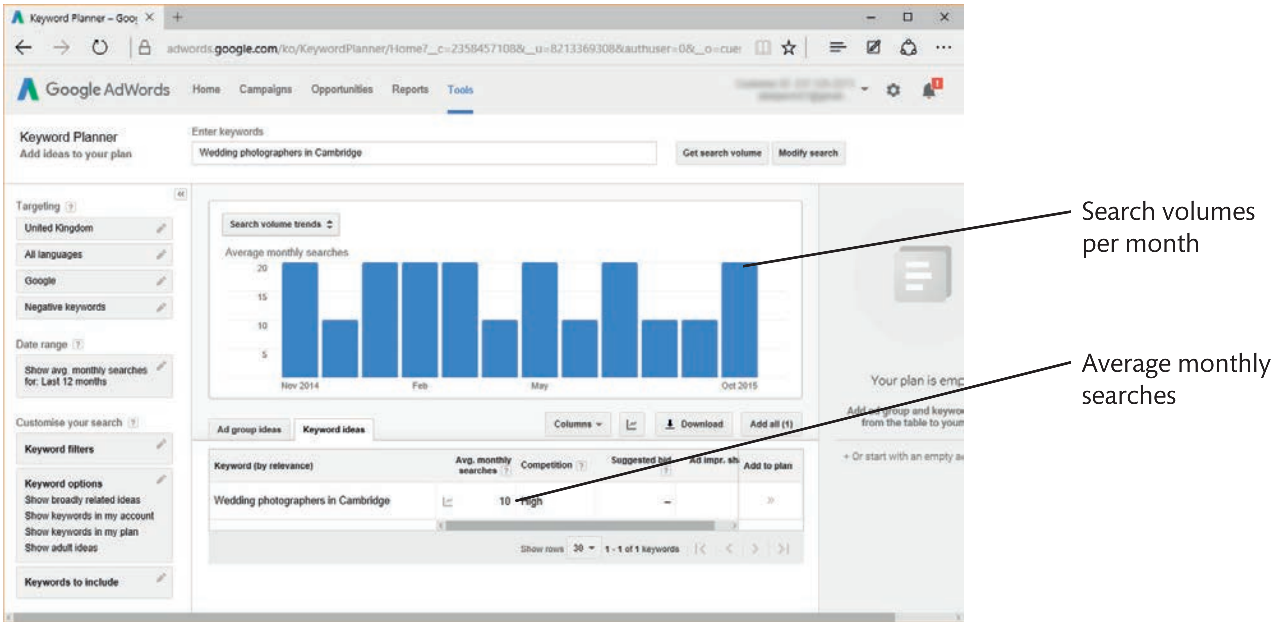
2 Go to the main page of Google Adwords™ at **www.google.co.uk/adwords/**



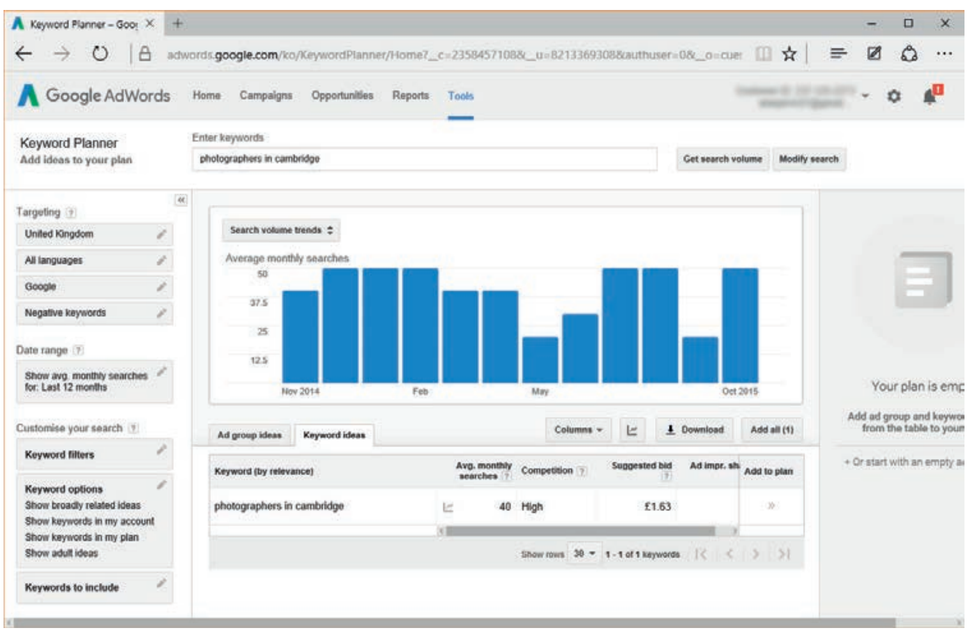
3 From the main page, choose Tools, then Keyword planner, which will then display the Keyword planner main page.



4 Choose **Search Volume and Trends** then enter a search string, for example 'Wedding photographers in Cambridge', then click **Get search volume** and you will see the results.



5 However, if you look at the volume and trends for 'photographers in Cambridge', you can see that the results are a little different, with an average of 40 searches per month compared with only 10 for 'Wedding photographers in Cambridge'.



6 Being able to do this kind of research, and finding out exactly what people search for, is invaluable to an organisation in helping them decide what keywords they should use on their website and in their social media profiles.

Timing

Establishing the best time to post content is another important consideration for anyone planning an organisational social media campaign. Social media is very dynamic, and if content is posted when your audience is not online they will most likely miss your post. An organisation, therefore, needs to identify those days and times when their target audience is most likely to be looking at social media. Once an organisation has sufficient people interacting with, for example, their Facebook page, then tools such as Facebook Insights will provide this information. If an organisation is starting from scratch, then research suggests that, for Facebook, on Fridays or at the weekend in the early afternoon are when most people are on the site, whereas Twitter™ is more popular during the week. Late at night or early in the morning is when fewest people are using social media. However, if the organisation sells a product internationally, then different time zones will need to be considered.

The advertising of some products is suited for posting at a particular time of day. For example, for a pizza takeaway restaurant, the best time to post on social media would be just before lunch or dinner time when people are likely to be hungry and thinking about what they should eat. Some organisations sell products which have a seasonal variety. For example, a garden centre might post articles about planting bulbs in Autumn, lawn mower reviews in Spring and information on how to water your garden in Summer.

Publishing schedule

The organisation also needs to consider how often they should post material to social media. Posting too often can be annoying for followers, but not posting enough can lose your engagement with the online community you had developed with your target audience. Research suggests that posting once a day on Facebook is about the right amount, while on Twitter™ approximately three tweets per day is best.

As part of the planning process for a social media campaign, an organisation should produce a publishing or posting schedule to show what content it intends to post and when (see Figure 8.5).

Social Media Content Schedule

Name: ....

Business: ....

Date: ....

	Date (Frequency of posting should be defined in your social media marketing plan.)	**Channels** (Enter all or name specific channels.)	**Type of post** (Sales, information, general interest, amusing.)	**Content** (Photo, video, web link, poll, questions, etc.)	**Comments**
1					
2					
3					
4					
5					
6					
7					
8					

Figure 8.5: An example of a social media schedule

PAUSE POINT

Being able to develop plans with timescales and targets is an important skill in many areas, and social media is no exception. Due to the dynamic nature of social media, plans need to be flexible and relatively short term (months rather than years). Set yourself five SMART targets for a social media plan for an organisation of your choice.

Hint SMART objectives need to be Specific, Measurable, Achievable, Realistic and Time-bound.

Extend Develop your social media plan further by outlining the target audience, what keywords you would use and how you would time your posts.

Developing an online community

There are a number of techniques that an organisation can use to help develop an online social media community. Posting content that will be of interest to the organisation's target audience and avoiding the 'hard sell' approach has already been mentioned, but there are many other promotional techniques that can be used.

People love to be asked their opinion, so asking questions, surveying opinions and requesting feedback is always a good way to engage with people. For example, a cake-making business could post 'our new flavour of cake is carrot and beetroot', but a more engaging post might be 'our new flavour of cake is carrot and beetroot, what do you think?' As well as engaging with the original post, people also like to respond to the comments left by other people. Perhaps an even better approach might be 'we are thinking of launching some new flavours, vote for your favourite', along with a Facebook poll. Another possibility is to link social media engagement with special offers, for example by offering a discount for customers who 'like' the organisation's Facebook page.

Of course, there is no point in asking your audience questions if you do not respond to them, so, to develop an online community, you need to monitor all your social media streams frequently and respond to any queries or requests promptly. If users post complaints or negative comments, then it is even more crucial that these comments are dealt with promptly and effectively.

Social media is not the appropriate place for detailed organisational and product information. This kind of information should be found on the organisation's website. Social media posts should, where appropriate, provide links to the organisation's website. This allows people who are interested in finding out more to find it on your organisation's website, but avoids having to include a lot of information in a social media post that many people will not be interested in.

Developing a social media policy

As already mentioned, there are a number of risks and issues associated with the business use of social media for organisations. One way to help reduce some of the potential problems is to create a social media policy. This document lays down guidelines for the use of social media within an organisation and anyone in the organisation who is associated with the use of social media should be aware of the contents of the policy. Some of the things which should be included in a social media policy are detailed below.

Organisational image and philosophy

It is important that the way an organisation presents itself through social media reflects the image that the organisation would like the public to have of it, and this should be outlined in their social media policy. For example, a cut-price supermarket would be happy to post content about special offers and price comparisons with other

supermarkets, whereas a high-end supermarket would not want to post comparisons with cut-price supermarkets or give their customers the impression that they were cheap, because their image would be that of offering a touch of luxury. Some organisations also have particular philosophies that they want to put across to their target audience. For example, some organisations might want to demonstrate that they are environmentally friendly or that they have high ethical standards: the Co-operative Bank is one such example.

Guidelines for content

The social media policy needs to state clearly what kind of content is acceptable to post and what is not. This will include things like ensuring that content is not offensive to anyone, and that it does not discriminate against anyone. It should use non-gender-specific terms (for example, police officer rather than policeman) and ensure that images show a mix of races. It may also cover guidance for staff such as never entering into arguments with customers, showing respect for everyone's opinions, not making negative comments about competitors' products and ways of dealing with complaints or negative comments. The guidelines will also normally encourage people to be open, honest and respectful in their communications, and to never tell lies even if the truth is not what the customer wants to hear. For example, being honest about prices, delivery timescales or mistakes that have been made is encouraged.

Guidelines for confidentiality

There is likely to be information that every organisation wishes to remain confidential, at least for a specific period of time, and this information should not be included in social media posts or in replies to customer's comments. This may include a whole range of things such as details of new products that are currently in development, phone numbers of staff members, details of why products are priced a certain way and the profit margins that the organisation intends to make. The policy should make it clear to staff what information is confidential and should not to be made public.

Guidelines for security

As the security risks involved in using social media are quite high, the social media policy will need to contain information about how to keep account details secure, how to avoid malware infections and other security issues.

Link

To remind yourself of the security risks involved for organisations using social media, see 'Increased vulnerability to cyber criminals'.

Separating organisation and personal content

Most of the employees within an organisation will have their own personal social media profiles and it is very important that they do not confuse these with those they are managing for their employers. The social media policy will need to remind employees that they must never use the organisation's social media streams for personal messages or to get too personal in their interactions with customers. It may also restrict what the employees can say about their organisation on their own personal social media profiles. In addition, some of these rules and guidelines may be included in the contracts of employment that employees sign when they join the organisation.

Legal and ethical considerations

There are a number of legal and ethical considerations that a social media policy should cover. The use of tracking cookies to enable personalised or targeted adverts, and complying with copyright laws when posting images and other content that has been created by others are just two such issues. EU law requires the consent of the user before cookies can be stored on their device. You will probably have seen these message pop-ups when you first visit a website, asking you if you consent to cookies being stored. Images and videos help to make posts more engaging but, unless the assets are created in-house, the copyright on the assets needs to be checked: that is, the ownership needs to be checked and permission sought for use of those assets on an organisation's social media pages and this might require payment of a fee. The social media policy may well forbid certain practices which are considered unethical, such as using so-called **'Black Hat' SEO techniques**.

Key term

Black Hat SEO techniques – techniques that attempt to fool search engines into ranking a website higher than it would have otherwise been, by violating the search engines' terms of service and possibly by creating fake customer reviews.

Reviewing and refining plans

Planning the use of social media by an organisation is not a one-off event, but rather an ongoing process of review, feedback and refinement. By reviewing the quality, effectiveness and appropriateness of pages and posted content, an organisation can, over time, determine which approaches to social media posting are the most

beneficial, enabling them to focus their efforts on these approaches in the future. Social media consultants will work with their clients and other relevant stakeholders, including the client's in-house team, to review and refine social media plans and they will need to consider the following.

Gathering feedback from a client and potential users

To review and refine social media plans, the main technique will be to gather feedback from clients and potential users (that is, the target audience). Some of the feedback from users will come in the form of analysis from a social media analytic tool such as Facebook Insights. It will also be useful to look at whether the organisation has achieved the targets it set itself for interaction on its various social media streams and whether it has met the success criteria that it set.

Communicating with a client

Sometimes, an organisation will be managing its own social media plans, in which case there would not be a client as such, only senior management that the social media team would need to report to. Larger organisations may hire a social media consultant to manage their social media plans, in which case representatives from the hiring organisation are said to be clients of the consultant. In such cases, the consultant would need to discuss their plans with the client in order to gain their approval and to improve the plans further. The hiring organisation may need to sign off on certain expenses involved in the social media plans. Working with a client can be difficult as they might not share your opinions or might disagree with your ideas. The important thing is to listen carefully to what the client says, and to consider what their aims are for their social media campaign. Remember that the client will probably understand the particular organisation much better than you do, but their understanding of social media may be very limited, so you need to listen to their explanations of how the organisation works and match that up with your knowledge of social media. Ultimately, the client is in charge because it is their organisation being represented on social media, but they will have hired a consultant for their expertise so should listen to their advice and suggestions.

Scheduling and documenting meetings

To work effectively and successfully with a client, you should have regular meetings with them. In these meetings, you (as social media consultant) should present your ideas and plans as they develop, and listen to the client's feedback. It is important to document these meetings (that is, to take notes) so that you know what was discussed, what feedback you received and any decisions that were made, in order to act on these decisions.

Agreeing and adjusting timescales

When planning a social media campaign, either in-house or with a social media consultant, it is important to agree timescales for when certain tasks need to be completed (such as setting up social media streams) and when targets are expected to be met. As mentioned above, communication is the key to a good relationship between a client and a social media consultant. Assuming that the communication is good, it would be fairly straightforward to agree to any changes to timescales within a social media plan based on the latest updates from the social media consultant on progress, or in response to changes within the client's organisation.

Refining ideas and solutions

Through gathering feedback and looking at whether success criteria and targets have been met, a social media consultant and client can see how successful their social media plans have been. They can also determine whether there are any parts of the plans, the targets they have set or the timescales they are working to that need to be refined or adjusted, in keeping with the organisation's business aims for their use of social media. The review work they have done, and good communication throughout, should enable the client organisation and the social media consultant to refine their plans and come up with any new solutions together, as well as reach any compromises that are required, and to agree a modified version of the social media plan.

Reflect

Working with social media will require you to respond to feedback from a number of sources, including the client and target audience (social media users who have liked/followed your social media pages). Receiving feedback requires that you do not take feedback personally, but in a professional and objective manner. It will also help you to learn from your mistakes, or at least improve upon your initial work, and to be flexible about your ideas. Reflect on how you would receive feedback in a professional and objective manner and respond to negative feedback about your work.

C Implement the use of social media in an organisation

Once you have created a social media plan, the next step is to implement it. You will need to select and use appropriate social media website tools and techniques to implement your social media plan.

Tip

You can practise using many of the tools and techniques required to implement a social media plan quite easily, because most social media websites will let you set up accounts and profiles for free, even if they are for an imaginary organisation.

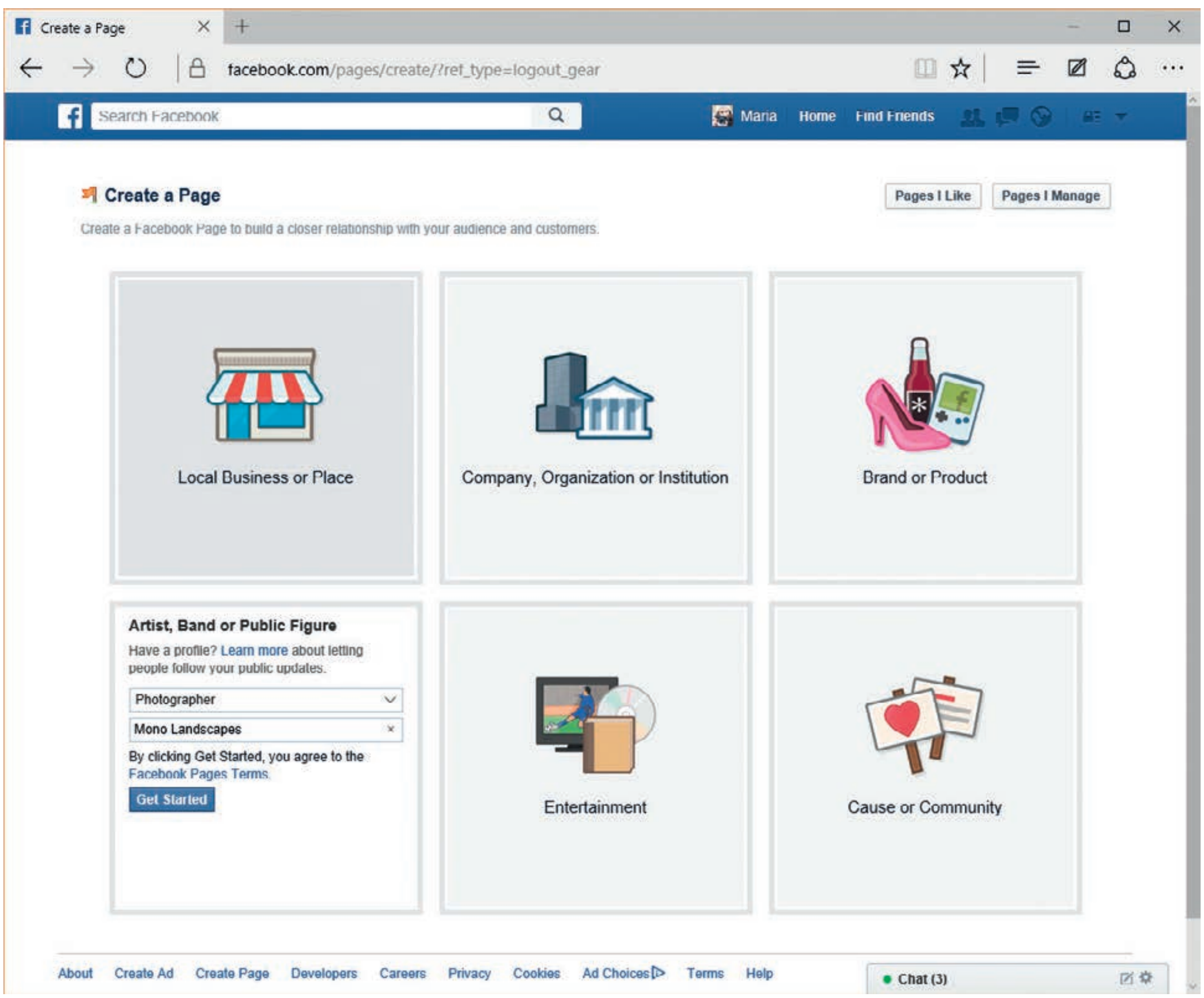
Creating accounts and profiles

Once you have identified which social media websites are to be used by an organisation, you need to sign up for accounts, create and set up profiles, and then perform ongoing administration.

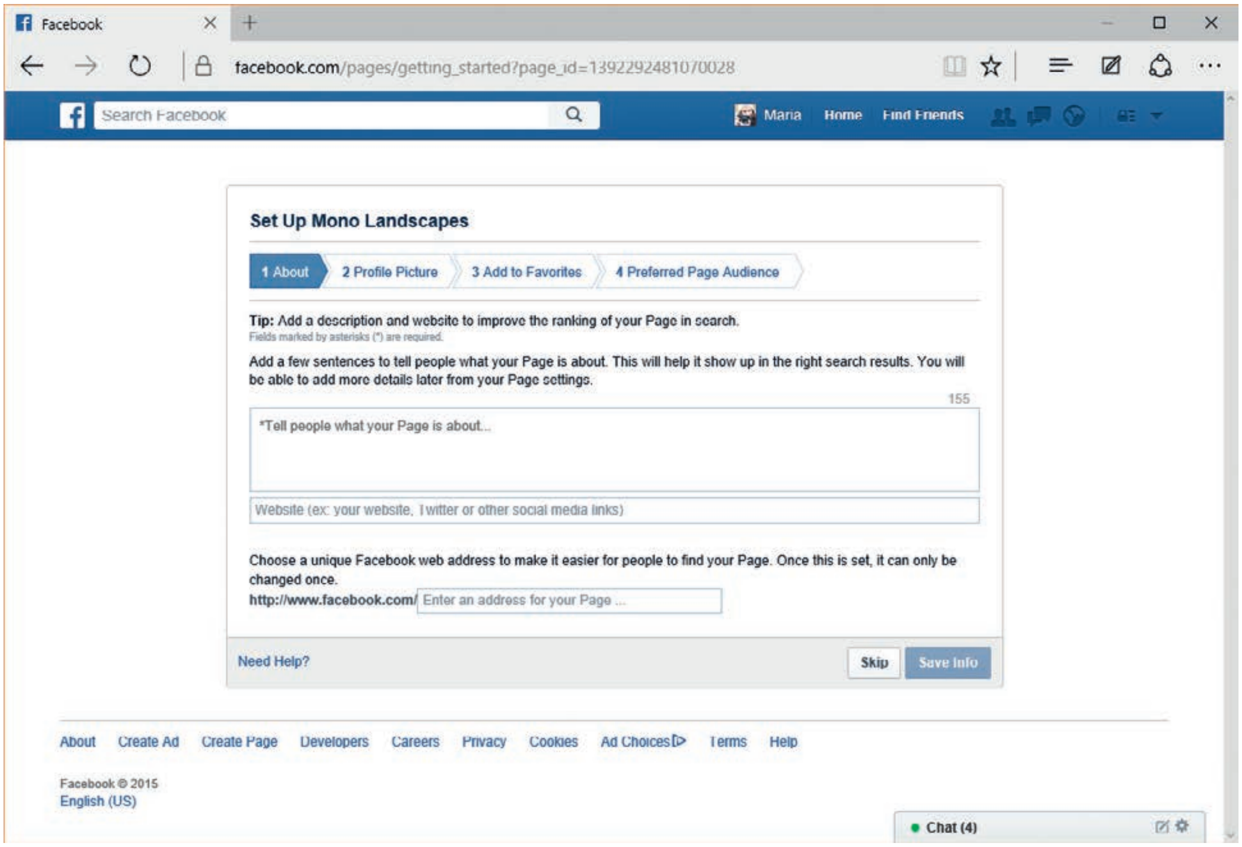
Step by step: Creating a page on Facebook

2 Steps

1 Facebook allows anyone with a personal page to set up any number of separate organisation pages. To do this, click the Create page option in the drop down menu that can be accessed via the down arrow at the top of the main Facebook newsfeed page.



2 You then need to select the type of page you want to create. You can choose from: Local Business or Place; Company, Organisation or Institution; Brand or Product; Artist, Band or Public Figure; Entertainment; Cause or Community. For example, if you wanted to create a page for a professional photographer, you could choose 'Artist, Band or Public Figure', then choose 'Photographer' from the drop down menu, enter the name of the photographer or the name of their organisation, and then click Get Started. Now you can work through the process of setting up the page.



Customisation and configuration of an organisation's profile

The organisation profiles created for the various different social media websites being used should all tie together in terms of the organisational image they are portraying. This is achieved through the design scheme used, tailored to each social media website's specifications, and through the text used to describe the organisation. They should all also include the keywords that have been identified for that organisation during the planning stage.

Most social media websites require the use of a background image or banner on profile pages. On Facebook, this is known as the cover image while on Twitter™ it is called the header photo. The profile page should also have an organisation logo, known as the profile picture on Facebook and profile photo on Twitter™.

You may need to put some thought into the cover image and profile picture. An existing organisation may already have a logo which can be used for the profile picture. A sole trader who offers some kind of service where they will meet their customers in person (such as a decorating or house cleaning service) might, alternatively, want to use an image of the person who runs the organisation and

delivers the service, to provide a personal touch. The cover image should represent the organisation and what it does in some way, or reflect the organisation's philosophy.

Tip

The area provided for the cover image is long and thin, whereas most images taken with a digital camera are much closer to being square, so you may need to edit a photo to make it work well in the space. However, you can reposition which part of the photo appears on the cover once you have uploaded it. Given the shape of the space available, only images which fit across a thin, horizontal area will work well; for example a vertical full-length image of a person would not be suitable as a cover image.

Figure 8.6 shows examples of profile pictures and cover images. Remember that, in order to build brand identity and recognition (so that members of the public recognise your organisation), you should use the same photos, colour schemes and text on all of the social media streams that the organisation is using (that is, your pages should all follow the same organisational branding guidelines).

When setting up your Facebook page, you should enter as much information about the organisation as you can, using the About page (see Figure 8.6). The profile About page helps people find your organisation. For example, if you enter a valid postal address for the organisation, then it will appear on Facebook Places and when people do a geographical search. Your organisation's profile is important because the text that you include in the short description is visible to anyone viewing your page, along with the address of your website (which you should also fill in on the About page).

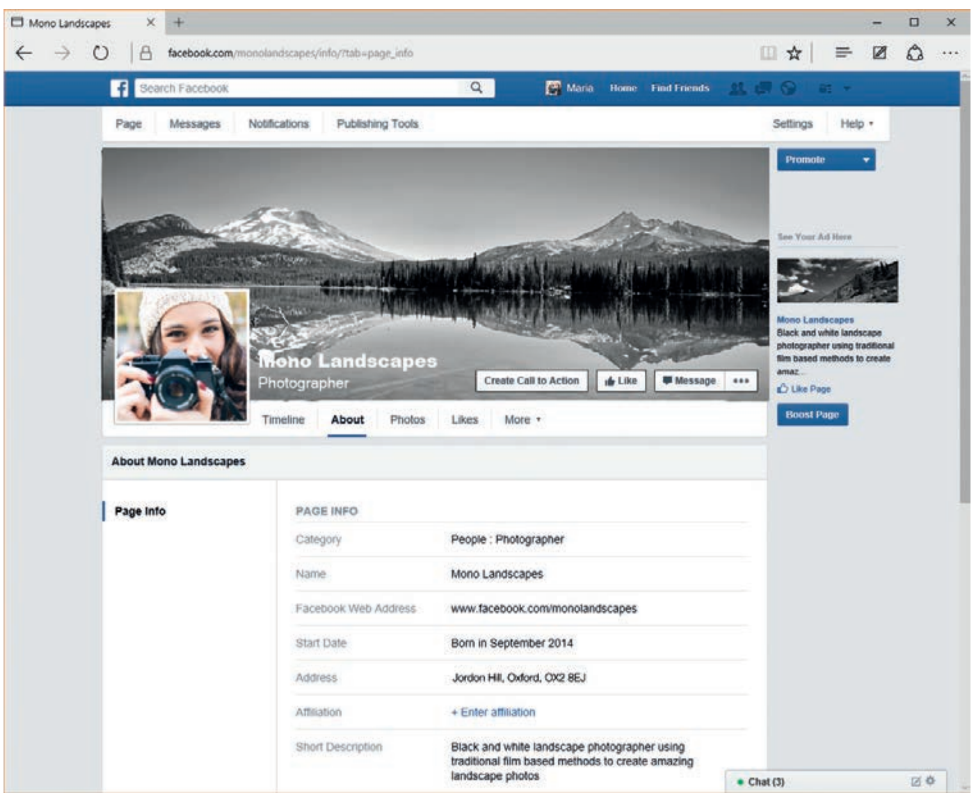


Figure 8.6: Facebook About page

The Settings page is also important; you can find this by clicking the Settings link at the top right-hand side of the page (see Figure 8.7). This controls many things about how your page works. The General settings, for example, allow you to develop a page, but keep it unpublished until you are ready for everyone to see it. You can also choose whether people can message you, if visitors are allowed to post to the page and a number of other things, including privacy settings.

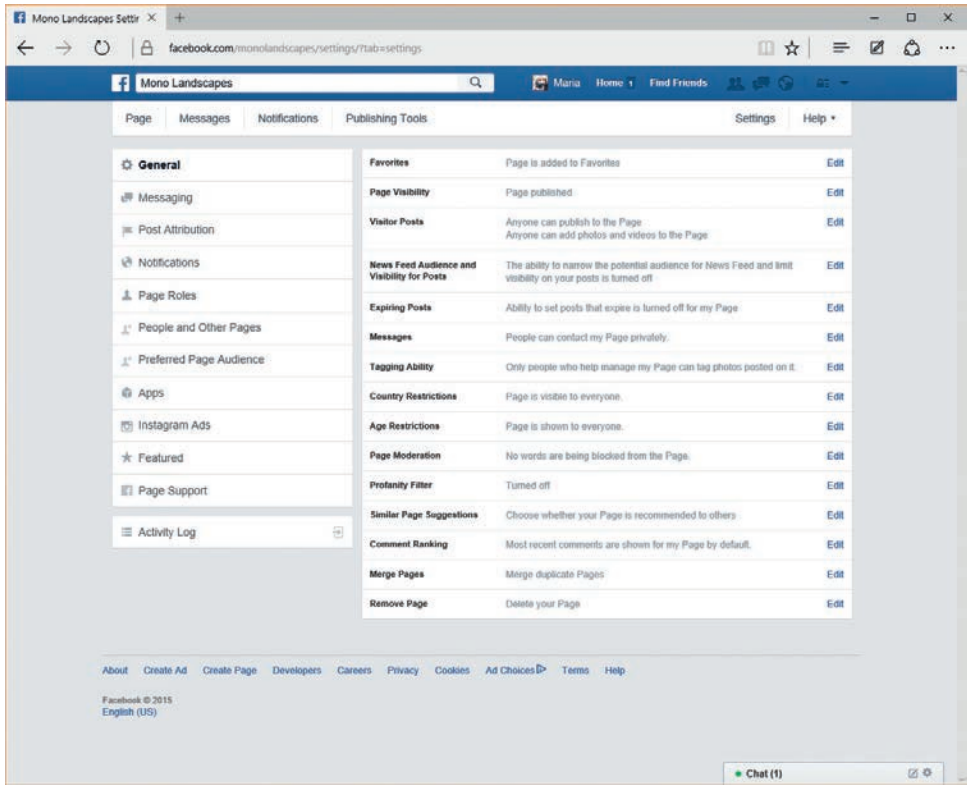


Figure 8.7: Facebook Settings page

Another very useful feature for an organisation's pages is Page Roles, which you can set up via the Settings page. This facility allows other people to run the page and you can give them the ability to do various things such as edit the page, post content to the page and view the page's Facebook Insights account. This means that, within an organisation, different people can have different responsibilities with respect to social media. For example, you could have one person responsible for monitoring and responding to comments/messages and another person responsible for posting new content to the page. To set up another person who can edit the page you need to click the Page Roles link on the left of the Settings page. You will then see the display shown in Figure 8.8.

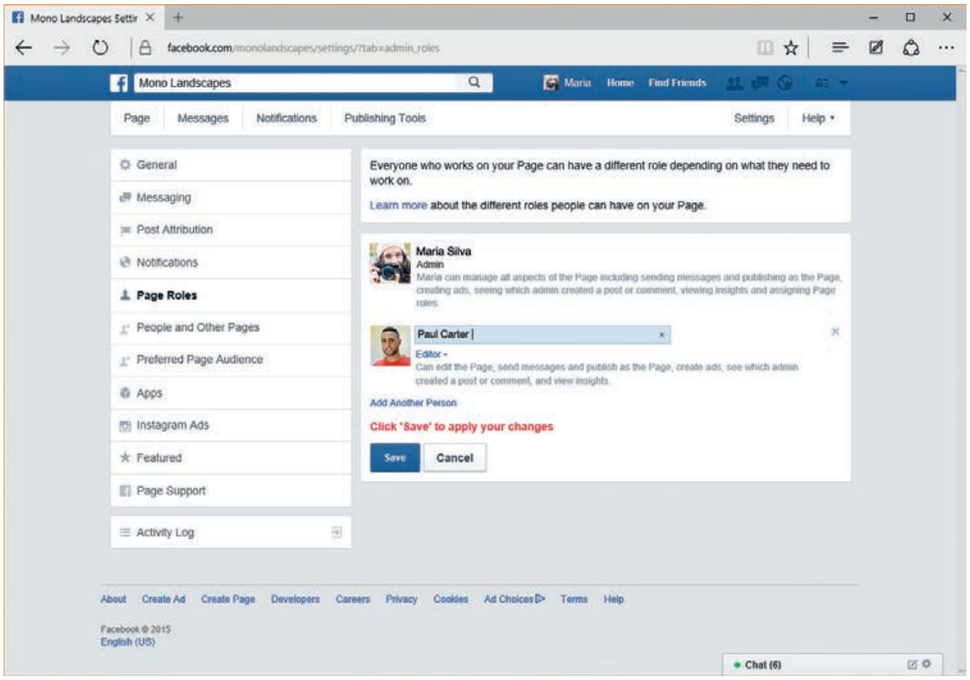


Figure 8.8: In this example, another Facebook user is about to be added as an editor for the page

An interesting feature of Facebook pages (introduced in February 2015) is the ability to set the Preferred Page Audience for your page. Anyone is still allowed to search for the page, but the page is more likely to be seen by people who match your target audience selections. Figure 8.9 shows the Preferred Page Audience settings.

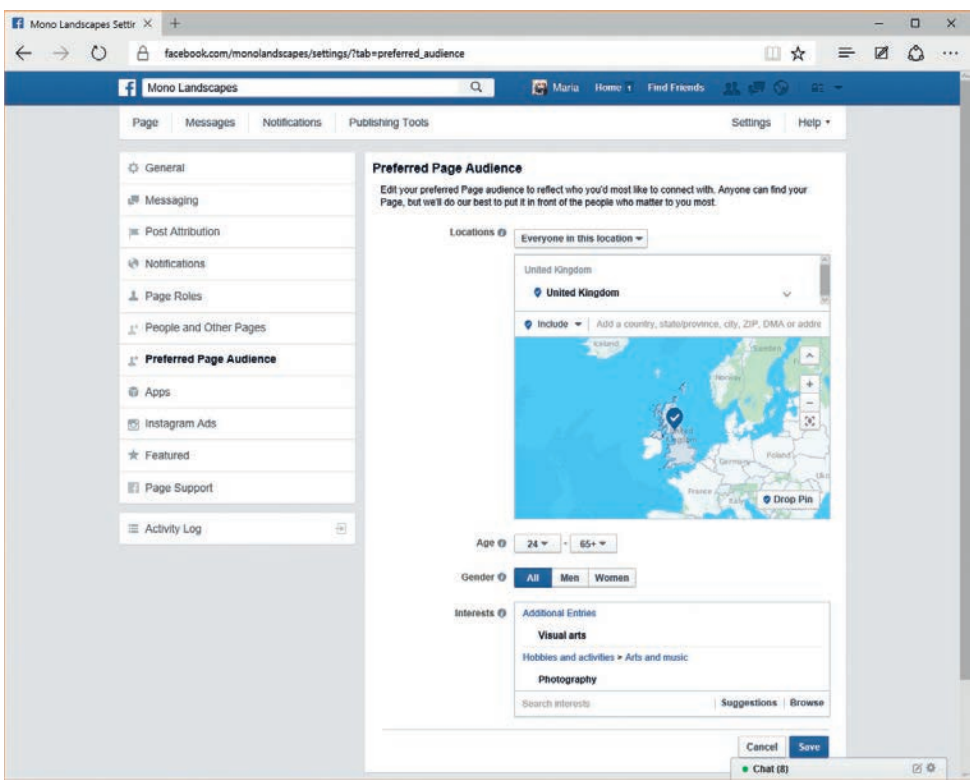


Figure 8.9: In this example, the Preferred Page Audience of a Facebook page has been set to people in the UK (you can add as many countries as you like)

There are a number of other useful things you can control on the Settings page. For example, the Notifications option allows you to set up email or text message notifications when a variety of things happen on your page. For example, you could receive an email every time anyone comments on one of your posts or every time the page receives a message.

Content creation and publication

During the planning stage, an outline publishing schedule and social media plan should have been created. Now is the time to implement that plan but it is likely, due to the dynamic nature of social media, that the plan will need to be changed and adapted as a result of how the public responds to it and to current events. For example, if an organisation sells a product which is related to the weather (such as ice cream or umbrellas), then they might need to adjust their posting schedule depending on changes in the weather.

Posts which contain images are likely to be more engaging than those that just contain text so, as part of the content creation process, you will need to obtain suitable images. Photos of the products that the organisation sells should be used, or photos that reflect the service that the organisation offers. Also, photos that tie in with the organisation's image and branding will be useful. Photos need to be of high quality as poor quality photos will reflect badly on the products/services and should, therefore, be avoided. Remember that, if you use photos that you have not taken yourself, you need to check the copyright situation and get permission to use them.

Carry out research to produce engaging content

You will also need to think about and research what kind of content will engage your target audience. There are two ways you can do this. You can carry out some market research by finding members of your target audience, showing them some content ideas and asking them which ones appeal to them most and why. However, this can be quite time consuming. The other way is to use social media analysis tools such as Facebook Insights and look at how much interaction each post generates. You can then modify the kind of content you post, based on the analysis you receive.

Improving visibility of published content

There is little point in putting a lot of effort into creating content and posting it on social media if no one sees it. Therefore it is important that you post at the right times on the right days when the majority of your target audience is online. This is something that you can check with Facebook Insights, as explained later in the chapter. There are also certain specific things you should do when posting on Facebook and Twitter™ to make sure that your posts are seen by as many people as possible.

Facebook does not show an individual user every post from all of their friends and the pages that they have liked. Instead, it uses sophisticated algorithms to prioritise those posts it thinks will be the most interesting to them. This is based on research Facebook has done into what users like to see and do not like to see in their timeline. Understanding how Facebook does this will help improve the visibility of the posts you make.

- One important factor that Facebook uses to decide if your post should appear on an individual's newsfeed is the affinity between you (your organisation) and that person. This is the amount of interaction (likes, comments and shares) between you and that individual. Therefore, whenever an individual comments on one of your posts, you should respond to help build up the affinity between you and that person.
- Try to create posts which encourage people to comment on them. One way of doing this is to ask questions rather than make statements. This will also help to build up the affinity between you and your audience.
- Post about relevant and trending topics and add links to relevant content from other websites or industry experts. For example, if you are a photographer and Canon™ launch a new camera with some special features, comment on this and add a link to the relevant page on the Canon™ website. The Facebook algorithms see this kind of posting as interesting to your audience so there is a greater likelihood that your posts will be widely seen.
- Avoid hard selling. Posts which simply encourage people to buy your products/services are not considered to be interesting so these types of post are least likely to appear on your audience's newsfeeds.

Research

The way Facebook selects which post to show is, as you might imagine, quite a hot topic in the social media world. The algorithm Facebook uses is often called 'EdgeRank' and an internet search for 'Facebook EdgeRank' will produce a lot of articles on the subject. However, ensure that you read the most recent ones as Facebook develops and refines the algorithm all the time.

To improve the visibility of your tweets on Twitter™, you should consider doing the following.

- When replying to a Twitter™ follower who has made a comment, it is common to start your tweet with an @ sign and the username of the person you are replying to e.g. '@bertsmith Thanks for that Bert we will look into it and let you know'.
- These types of reply are not seen by all your Twitter™ followers, only by people who follow both you and the person you are following. If you want all your followers to see the tweet, then do not put the @username at the very start of the tweet.

Another way you can make sure that your posts are seen by your target audience is to use paid-for posts. An organisation would need to weigh up the cost of a paid advert campaign against the time and effort required to run a 'free' campaign. The two are not mutually exclusive, of course, and an organisation might well use a paid-for campaign to get their social media presence 'off the ground' and then maintain and develop the followers with a 'free' campaign.

Tip

It is very important to proofread posts before you actually post them. Posts with spelling mistakes or other errors will reflect very badly on the organisation. If possible, get someone else to check the posts (because it is much harder to spot mistakes in your own work), not just for spelling errors but also to ensure that the meaning of the post cannot be misunderstood and that it is not likely to cause offence to anyone.

Encouraging audience interaction

- The importance of encouraging audience participation has already been mentioned. Posts need to be phrased in a way that is entertaining and focused on your target audience without being pushy. Including images in posts is always a good idea as people respond better to images rather than text alone. The use of the **click bait** approach is not recommended as many social media users find it annoying and the Facebook EdgeRank algorithm does not favour it. Encouraging comments by asking questions is one way; you can also ask people to post photos of themselves using your product. Other possibilities include using surveys (which are useful for obtaining feedback on new product ideas) and quizzes.

Key term

Click bait – a term used for content that encourages users to 'click through' the post to see the linked content on an external website. Click bait content often uses sensationalist headlines such as 'You won't believe what happens next' to exploit the viewer's curiosity.

Integration of information across the organisation's website and social media profiles

- For organisations that sell their products through a website, one of the main goals of using social media will be to encourage people to visit their website and, ultimately, to purchase their products. Conversely, they will also want people who visit their website to like their Facebook page and follow them on Twitter™ and other social media websites. Therefore, an organisation's social media pages and their website should, as far as possible, be integrated in a number of ways.
- The organisation's website should feature social media buttons to encourage visitors to like/follow their social media pages.
- The formatting of both the organisation's website and their social media pages should reflect a consistent organisational branding style. For example, the background photos and profile images used on the social media page should match/reflect the images and colour scheme used on the organisation's website.
- Social media posts should, where relevant, contain links to the organisation's website. For example, when a new product is launched, brief details can be given in a social media post, with a link to the product page on the organisation's website, where much more detailed information on the product can be found.
- Overall, the idea is that an organisation's social media pages and their website should integrate with each other, and, of course, to develop brand identity, they should have a consistent look to them.

Adapting and testing content on different device platforms

- It is likely that you will develop and post content using a PC, but many of your target audience will view the content using mobile devices such as smartphones, tablets or notebooks. You can check which devices and web browsers your audience uses with analysis tools such as Facebook Insights. It is a good idea to check how your content looks on different devices, especially those with a much smaller screen, to ensure that your

Reflect

To work in the social media industry you will need good communication skills. Writing posts which are short but convey the intended meaning, and using the right tone and language, are real skills. You also need to be able to respond to social media messages, which might include negative comments and complaints, in a positive and helpful way. Setting the right tone when replying to these types of messages is not easy, but you should try to be positive and helpful. Think about techniques that you could use to write short and clear posts and how you would respond positively and helpfully to negative comments.

audience will see the posts in the way you intended them to. It is also worth bearing in mind that, as the account holder, the posts you make will often look different from how they appear in a user's newsfeed. For this reason, it is often useful to have a separate social media account available to help you to check how your posts appear to your audience.

Implementation of online community building

Ways to develop an organisation's online community have already been outlined. One approach to develop an online community further is to join other related online communities, rather than just staying within the organisation's own social media pages. This can be done in number of ways.

- Joining Facebook groups on relevant subjects, joining Google+™ communities or following appropriate accounts on Twitter™ can provide useful sources of information. For example, a photography business could join one of the many photography groups on Facebook or Google+™. While many groups frown on businesses posting advertising material, there is nothing to stop the business answering questions and making comments. Many bloggers tweet details of their latest blog posts and, by following relevant bloggers on Twitter™, organisations can keep up to date with interesting and relevant developments. This can also provide a good source of material for posting on your organisation's social media pages. Sharing (on Facebook) or retweeting (on Twitter™) interesting relevant posts from other people and organisations you have followed is a good way to develop your online community, and help develop the idea in people's minds that your organisation is knowledgeable and a useful source of relevant information.
- Tagging photos in Facebook is also a way of trying to encourage engagement. For example, suppose that a car dealer asks each person who buys a new car to have a photo taken with the car they purchase. They then ask the person if they would mind posting the picture on Facebook and tag the owner in the pictures (assuming they have a Facebook account). If they are happy, then the picture will not only go up on the organisation's Facebook page, but will also be seen by all the Facebook friends of the person who bought the car, thereby expanding the online community.
- Twitter™ hashtags provide a way to join in an online conversation about a specific topic. If an organisation sees a hashtag relevant to their organisation, then they can tweet a comment using the hashtag.

Link

Look at Developing an online community, for more on how to develop an online community.

All of these suggested ways of developing an online community are designed to raise the profile of the organisation in the minds of people that follow their social media pages. Rather than annoying the organisation's target audience with pushy direct selling posts, the idea is to build up an impression of an organisation that knows what it is talking about and is up to date with all the latest information and developments, making this the sort of organisation that people feel they can trust and can do business with.

Monitoring and responding to comments and automating content posting

- It can be difficult to keep track of all the different social media streams and to time postings at the optimum times. Fortunately, there are a number of different tools which can be used to help manage the task of running a social media campaign. One of the best known tools for this is called Hootsuite® and, as long as you only want to manage a maximum of three different social media streams, it is free to use.

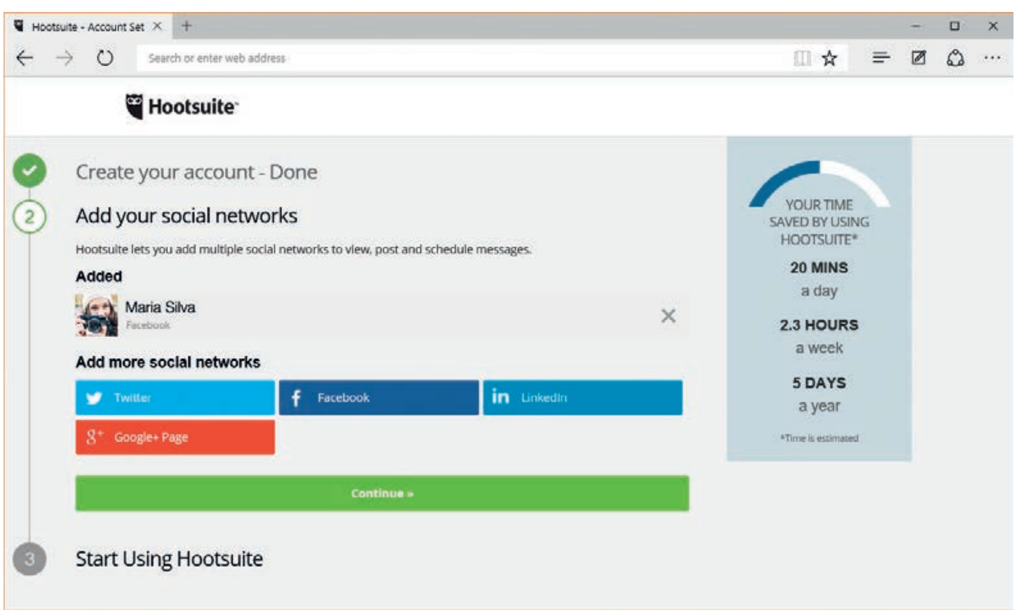
There are many other social media analysis and management tools available. Some of the best known free ones are:

- TweetReach – allows you to analyse how many times your tweets get retweeted
- Klout – uses a variety of measures to identify what people think of your brand and what kinds of post influence them the most
- Social Mention – allows you to monitor the use of different keywords across a wide range of social media websites
- TweetDeck – a Twitter™ management tool that allows you to schedule tweets, track hashtags and manage multiple accounts
- Google Alerts™ email update service – allows you to monitor the internet for any mentions of your brand, your competitors or any keyword you choose.

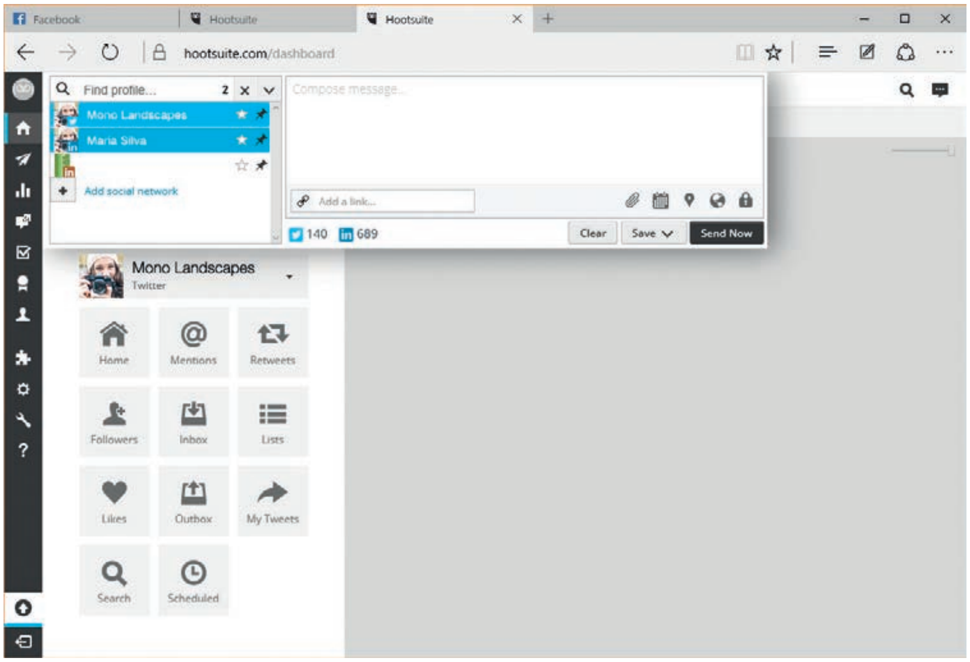
Step by Step: Hootsuite®

6 Steps

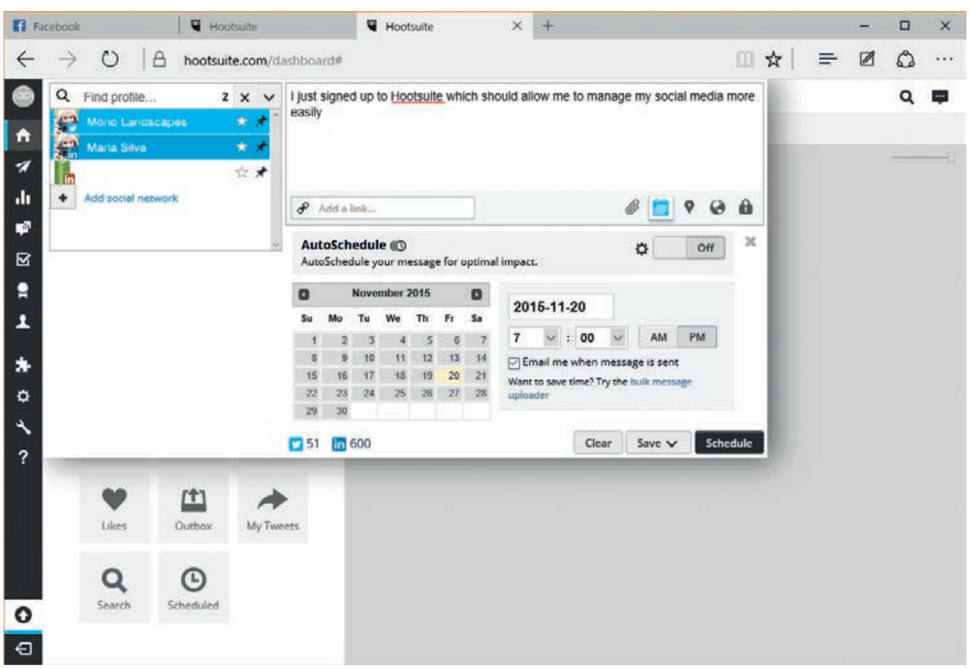
1 Here is a look at some of Hootsuite®'s features. If you log into Hootsuite® using your Facebook account you will see something like this.



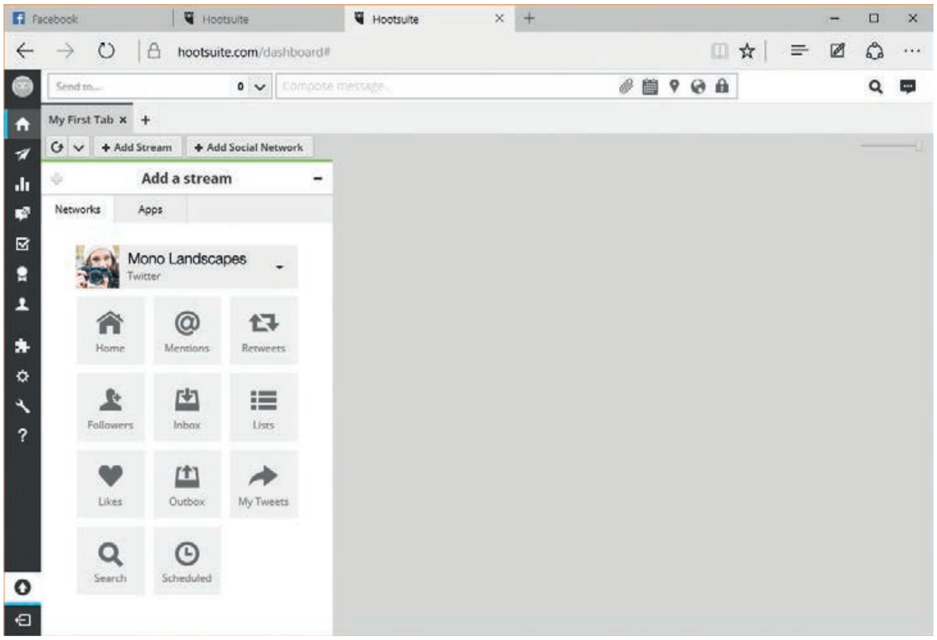
2 You then need to add your other social media pages using the buttons on the page. With the free version of Hootsuite®, you can only add three social media accounts. Hootsuite® has lots of features but one of the most useful is the ability to create a single message and have Hootsuite® post it for you on multiple accounts. When you open the main Hootsuite® dashboard the message creation area is at the top.



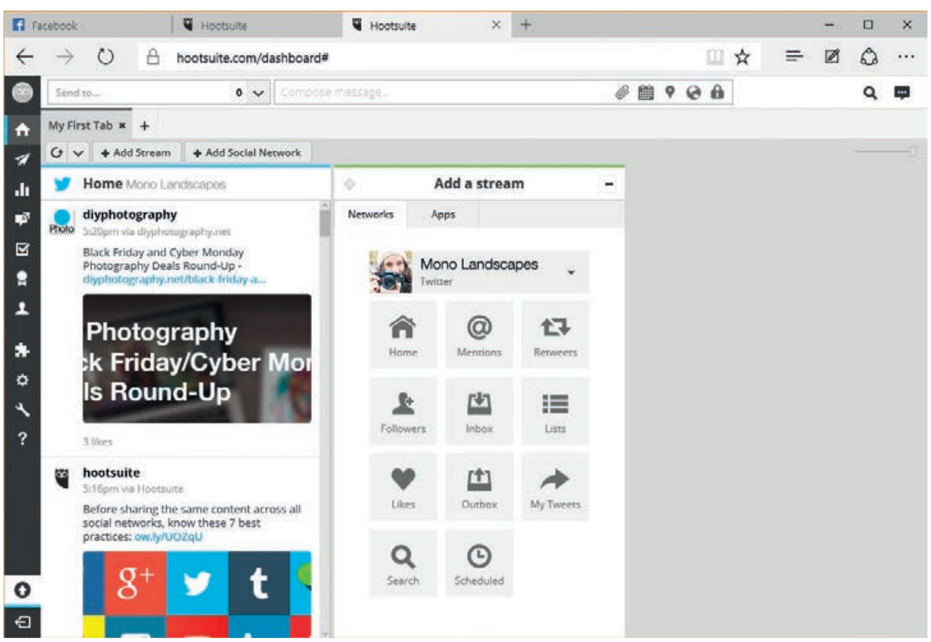
3 You can then type the text of the message you want to post and add links if required. The figure below shows both this and the scheduling option which allows you to choose a time when the post is made. You can, therefore, create posts at a time which is convenient to you but actually post them when your target audience are most likely to be online.



4 Another useful feature of Hootsuite is the ability to create social media 'streams' and display them on your Hootsuite® dashboard. This means that, rather than having to visit each social media page to see what is going on, you can have all the updates from your social media accounts shown in one place. To set this up, click on the Home button on the panel that runs down the left-hand side of the dashboard and the 'Add a stream box' should pop up.



5 Choose the social media account you want to add the stream for (Twitter™ shown here), and then select what aspect of the stream you want displayed (Home is a good starting choice). The figure below shows a Twitter™ stream added.



6 Further streams can be added to complete the dashboard and allow you to monitor multiple social media pages at once.

Data gathering and analysis

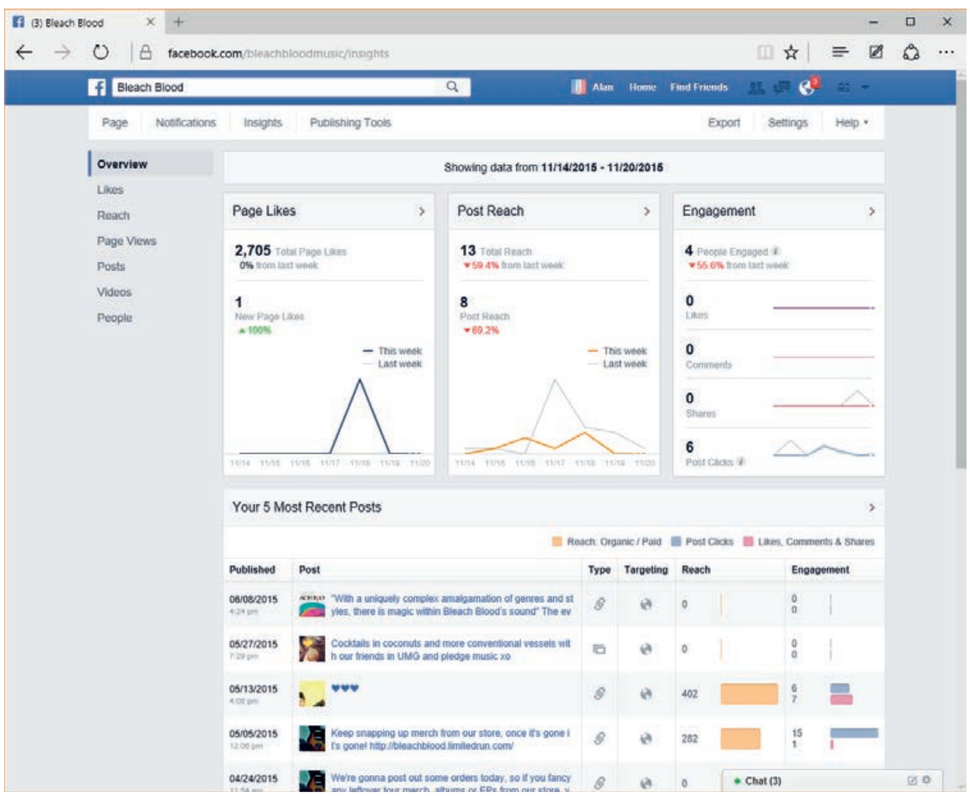
One of the major benefits of using social media over traditional methods of promotion is that tools are available to allow you to investigate how successful your posts have been and who your audience is. Facebook, for example, provides a free tool called Facebook Insights which provides detailed data on the effectiveness of your social media efforts. Twitter™ provides Twitter Analytics (also called the Twitter™ Activity Dashboard), which provides data on how many tweets you have made and how many people have seen them. There are also many other third-party tools available.

Google Analytics™ is a powerful tool for analysis of website traffic, but it does not provide any social media data. However, it is a useful tool if the aim of using social media is to drive customers to your organisation's website, as it will show the source of these visitors, so that you can see how effective your efforts have been and how many people have 'clicked through' your social media pages to your organisation's website.

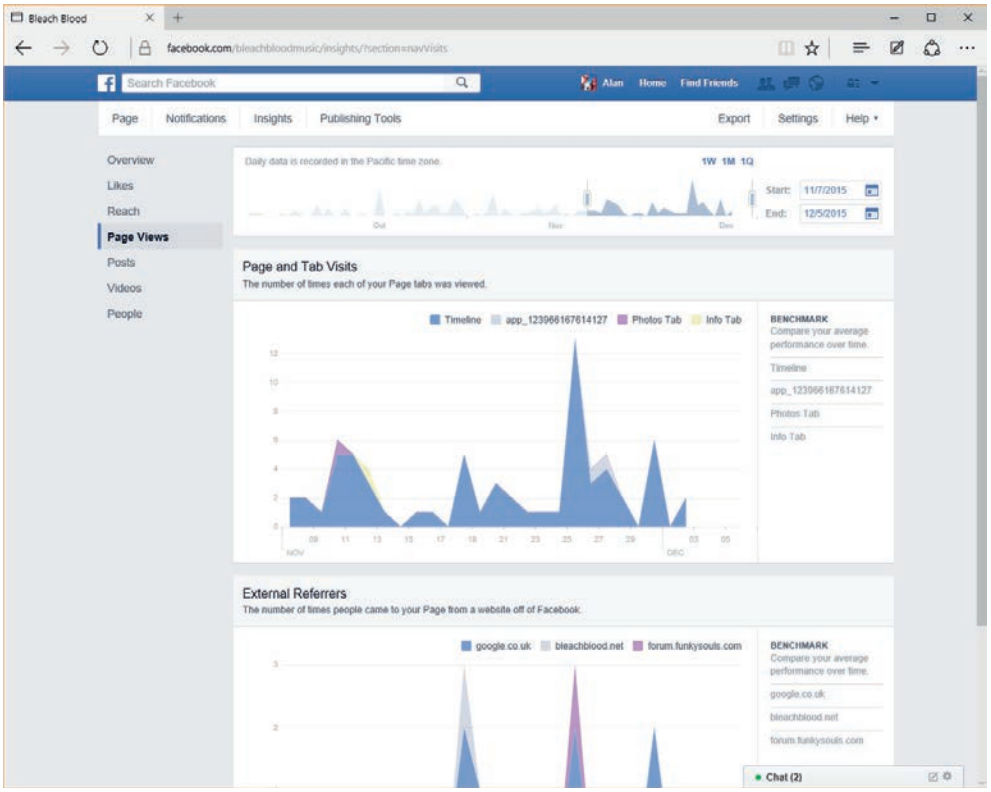
Step by step: Facebook Insights

4 Steps

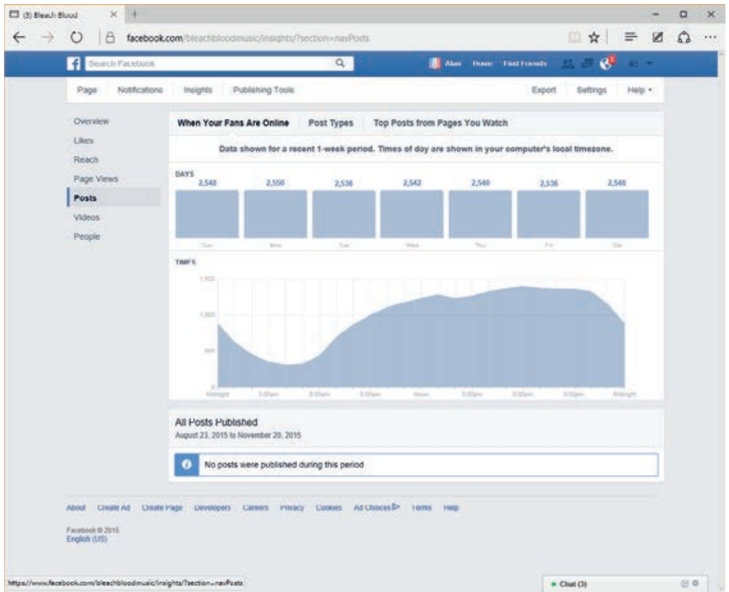
1 Any organisation's page can use Facebook Insights as long as it has more than 30 'likes'; you cannot use Insights on your personal Facebook page. The Facebook Insights link appears in the menu at the top of your Facebook page. Once you click on the link to Facebook Insights, the Overview page is shown.



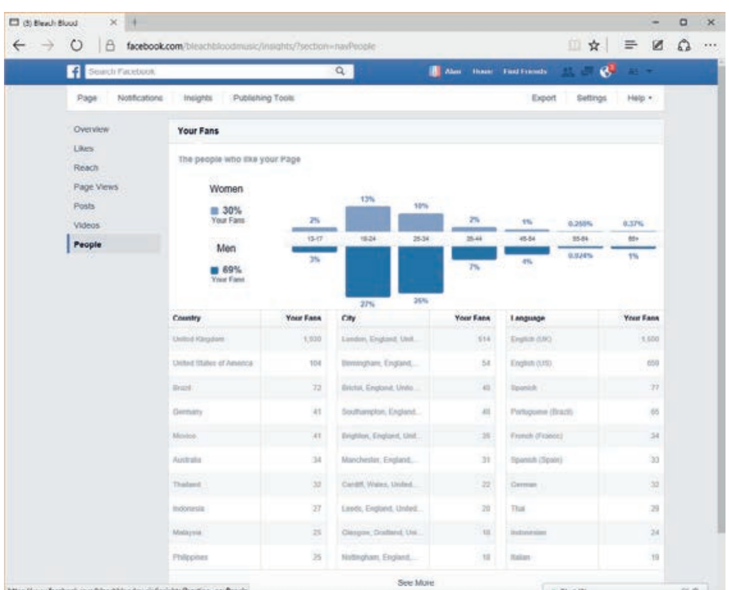
2 The Likes box shows the total number of likes the page has received and the number received in the last week. The **Post Reach** shows the number of people who have seen your posts in the last week, and the **Engagement** box shows the number of people who have interacted with the page. In the lower part of the page, is a list of information on the last five posts on the page. This is particularly useful as you can see how effective individual posts have been in engaging with your audience, and the charts on the right give a visual impression of the effectiveness of each post. A link at the bottom of the recent posts table allows you to expand the table to see all posts. Down the left-hand side is a menu from which you can choose to see different data about your page. For example, the Page Views link will display a graph showing how many people viewed your page. The figure here shows an example of this.



3 The default view shows you the page views over the last month, but you can change the period over which the views are shown on the graph at the top right-hand side of the page. This allows you to see the popularity of your organisation's page over time. Lower down on the page, it shows how many times people came to your Facebook page from other websites such as Google™ or your organisation's website. The Posts link will display charts which show when people who have liked your page are online. This information is very useful as it helps you to decide on the best time and day to post.



4 Another interesting page with lots of useful data is the People page. This provides data about the demographics of people who have liked your page (e.g. their age group, gender and location). In this figure, the page has a predominately male audience (69 per cent male compared with 30 per cent female) and 75 per cent of the people who liked the page are in the 18 to 34 age range (40 per cent are 18 to 24). There is a lot of other information that you can display with Facebook Insights. The Help menu, at the top right-hand side of the pages, contains information on how to use Facebook Insights.



Key terms

Post reach – the number of people who see a particular post.

Engagement – the number of people who interact with a post in some way (like, comment or share it). Engagement shows a stronger effect than Post reach. It means that people not only saw your post but found it interesting enough to interact with.

The data that tools such as Facebook Insights or Twitter Analytics provide can help you to develop and focus a social media campaign in a number of ways.

- It can identify how much interaction each post you make generates. Posts that generate interaction are what you should be aiming for, and the data should help you see the types of post that are most successful in this respect. This means that you can concentrate your future efforts into developing the types of post that generate a lot of interaction and waste less effort on those that do not.
- You can also identify your audience profile in terms of age, gender and location. Once you know this, you can compare your actual audience with your target audience. If there is a significant mismatch (for example, your actual audience is much older than your target audience or your page attracts more males than you were aiming to), you can do one of two things. Either adjust the content you are posting to try to attract more of your target audience (that is, post more material that might be of interest to a younger audience and/or female audience), or adjust the profile of your target audience to match the one you have attracted. (This second option might also involve rethinking your product or service and would have many added complications, but sometimes might be necessary.)
- The data will reveal the location of your audience. If you run a geographically-limited business (such as a dog grooming service), then having lots of followers in the US or Japan is not particularly beneficial (although you might want to consider the possibility of setting up a blog). Therefore, it might be worth making your posts more locally focused. If your organisation can deliver its products by post and you attract a large non-UK audience, you might want to consider encouraging overseas buyers, for example by mentioning overseas shipping prices.
- You can monitor the number of likes and shares. This is useful if you have set yourself a target of achieving a certain number of likes or shares during a specified period of time. It is also a way of monitoring the success of the content you post and identifying which types of post achieve the most likes or shares.
- A common concept in marketing is 'trans media story telling', where one campaign is stretched across multiple platforms, each with an element of the campaign that is not available on the other platforms – a bit like a digital treasure hunt.

PAUSE POINT

Tools such as Facebook Insights provide valuable objective data to help you to evaluate the success of your social media campaign and your progress towards the targets you have set. This can really help you understand what works well and what is not so successful, so that you can improve your own performance by focusing your efforts in areas you know have been successful in the past.

Prepare five posts, including a variety of content formats, for an organisation of your choice. Share these via a Facebook page that you have set up for this organisation. Get your classmates, friends and family to interact with the posts. Then use Facebook Insights to analyse how successful each post was in terms of interaction.

Hint Look back through learning aim C for ideas about the kinds of content you could create for posts, and about how to use Facebook Insights.

Extend Write a report on which post was most successful in terms of interaction and why you think this was. In your report, you should also discuss how your audience is composed in terms of demographics and location.

Case study

Blogging

If your social media pages develop a lot of followers, you might want to consider setting up a blog on a particular subject. Bloggers post articles at regular intervals (often daily) which are more detailed and in more depth than social media postings. The blog articles are usually promoted to their social media followers by posting a link to the latest blog article. Bloggers who are really successful and have lots of followers can make money from adverts placed beside their blog articles. However, becoming a successful blogger requires a lot of commitment in terms of the time needed to develop the articles and build your audience.

Search engine optimisation

Search engine optimisation (SEO) is important to any organisation, whether or not they are using social media. SEO is a set of techniques that can be used to help an organisation improve the search engine ranking for its website. People browsing the internet often only look at the first few links provided by a search engine so, if you want to ensure that your organisation gets noticed, when relevant search terms are used, you must ensure that your website is optimised for search engines.

Principles of search engine optimisation and links to social media

Unit 15: Website development covers SEO techniques in more detail. In this section, you will look specifically at how some principles of SEO relate to the use of social media. Organisations should integrate their websites and social media pages closely. It is very important that those involved in developing and implementing a social media presence for an organisation are aware of best practice SEO.

- Keyword research and strategy and social media profiles: All organisations should have a keyword strategy which identifies those words and phrases that people are most likely to use when searching for the organisation's products and services. The strategy should be implemented consistently across the organisation's website and all their social media accounts.
- Social media profiles and use of keywords in context: Your organisation's social media profiles should use the same keywords as your organisation's website. Moreover, when developing text content to post on your social media profiles, you should include your organisation's keywords, where possible.

Link

You can find out more detail about SEO techniques in *Unit 15: Website development.*

Link

Keywords, in the context of posted content, are discussed in 'Keywords and their use in posted content', in 'Organisational uses of social media for business purposes'.

- Website URL and content: Ideally your organisation's website URL should include its top (number one) keyword or phrase, so that your organisation's website will be included higher in the search engine rankings for that keyword or phrase.
- Your organisation's website should also be constructed in a search engine-friendly way. Ways to do this include using alt text for images, so that the search engine knows what is in the image, and using proper title tags to describe each web page.
- Significance of regular updates: Search engines will assume that websites that are not updated very often are unlikely to be useful to people, so it is important to update your organisation's website content regularly. Websites which are regularly updated will be ranked higher by search engines for this reason.
- Importance of inbound links and ways of developing them: When other websites link to your organisation's website, this is called an 'inbound link'. Search engines assume that if other websites think that your organisation's website is worth linking to, then it must have some relevant information on it and will therefore put it higher in the search engine's rankings. Developing inbound links is not easy as you have to persuade other organisations and businesses to include links to your website on their websites. However, you can encourage this by returning the favour.
- The best way to get inbound links is to have useful and relevant content on your website that other organisations want to link to. Social media can be useful in developing an online community of like-minded people and this can be helpful in getting people to link to your website.
- Use of social media to encourage visitors to the organisation website: You should ensure that many of your organisation's social media posts include links to relevant pages of your organisation's website. These social media posts should be used as brief 'tasters' of your organisation's products, services or promotions. By providing links to pages on your organisation's website in the social media posts, people can read the taster and then follow the link to find out more.
- Timescales for achieving change in search engine rankings: SEO techniques are not a one-off fix that will improve search engine rankings overnight. Search engines take time to develop 'trust' in a website and it will often take 3-6 months to see any real improvements. Also, search engines are always tweaking their search algorithms, so it is important to keep up to date on the latest advice for improving your SEO for specific search engines.
- Importance of search engine rankings linked to a social media service (Google+™): As mentioned earlier in the unit, one of the main reasons for having and using a Google+™ social media account is that is can help to improve your search engine rankings in Google™'s search engine.

Monitoring website interaction

Tools such as Google Analytics™ can tell you a great deal about how people use your website: how people get to your website, how long they stay on it and which web pages they visit while they are there. This data can help you to optimise your website and therefore make it more appealing and useful for visitors. The same is true for similar tools, such as Facebook Insights, which monitor your social media accounts.

For example, the data will identify pages on your organisation's website that people do not stay on for very long. This will allow you to consider why people do not stay long on a particular web page and how you might encourage them to stay on it longer. Knowing how people get to your website is also useful. For example, did they come directly from a search engine (Google™ for instance), or did they follow a link from your organisation's social media posts or from another website which links to yours?

Understanding how people interact with your organisation's website and your social media accounts will help you optimise them, because it helps you to better understand your audience and allows you to make amendments to the pages to make them more appealing, relevant or usable.

Assessment practice 8.2

B.P3 B.P4 C.P5 C.P6 C.P7 B.M2 C.M3 BC.D2 BC.D3

One of the small-business people you gave a presentation to for Assessment practice 8.1 has asked you to develop a social media campaign to promote their photography business. You need to:

- produce a plan for using social media to promote the business, including a keyword strategy and consideration of SEO, justifying the choices you make and showing how they will meet the purpose and business requirements
- review the plan with others to refine and improve it
- produce appropriate social media content for the business in line with the requirements of your plan
- review data on social media usage and interaction and optimise the content and format/features of the social media website in use
- write an evaluation of the plan you created and its implementation that considers how well the business requirements were met
- demonstrate individual responsibility, creativity, and effective self-management in the planning and use of social media in the context of the photography business.

Plan

- What is the task? What am I being asked to do?
- How confident do I feel in my own ability to complete this task? Are there any areas I think I may struggle with?

Do

- I know what it is I am doing and what I want to achieve.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.

Review

- I can explain what the task was and how I approached the task.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

Further reading and resources

Websites

There are many blogs and websites related to social media. Some of the best known are:

- The Social Media examiner: **www.socialmediaexaminer.com**
- Buffer Social: **blog.bufferapp.com**
- Razor Social: **www.razorsocial.com/blog**
- Post planner: **www.postplanner.com/blog**

Books

There have also been many books written on the subject. It is always wise to check the publication date of a book in such a dynamic area, as books more than a few years old are not likely to be relevant.

Kitchen T, and Ivanescu Y – *Profitable Social Media Marketing: How To Grow Your Business Using Facebook, Twitter, Instagram, LinkedIn And More* (CreateSpace Independent Publishing Platform, 2015). ISBN 9781519611925

Ryan D – Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation (Kogan Page, 2014) ISBN 9780749471026

Ryan D – *Understanding Social Media: How to Create a Plan for Your Business that Works* (Kogan Page, 2015) ISBN 9780749473563

Steam A – *Make Social Media Work for your Business* (CreateSpace Independent Publishing Platform, 2014) ISBN 9781502911490

Brown MJ – *Social Media Marketing, 2nd Edition* (CreateSpace Independent Publishing Platform, 2015) ISBN 9781512088298

THINK FUTURE

Simon Jones

First year journalism student

After school, I took a Media degree course as I found the subject interesting and I wanted to work in journalism. During my first year, we have completed modules in media photography, radio production and new media (which includes social media). The course has been hard work but I have enjoyed many parts of it and learnt a lot of new things. I know, however that this is a really competitive area and, to progress into a job in media, I am going to have to demonstrate excellent technical skills and creativity. I had a great opportunity to develop some of my skills with a holiday job in my uncle's builders' merchant business recently. It's a medium size organisation with five large builders' yards in the Wolverhampton area. He asked me to develop the business's social media presence as they only had a Facebook page and it hardly ever got updated. Work experience like this was a great opportunity for me as it allowed me to develop real-world skills to add to my academic studies.

Focusing your skills

Planning a social media presence

To be able to develop a social media strategy for the builders' merchant there are a number of things to consider.

- How well do you understand the organisation? Spend some time in the yards watching how the staff work and talking to customers to understand how social media might be able to work for the organisation.
- What does the owner of the organisation want to achieve by using social media? Is it realistic and achievable?
- What timescales are you working to?
- How will the staff be trained in the organisation's social media procedures?

Where to start? Once you have learnt as much as you can about the organisation, you need to formulate an outline social media plan covering the following.

- **The purpose of using social media in the organisation.** For example, you might decide that the main purpose is customer service, providing a method by which customers can enquire about stock levels. A secondary purpose might be to provide information to customers about new products and special offers.
- **The social media websites you will initially use and why.** For example, you might decide to use Facebook and Twitter™ as they provide the messaging facilities that you need to provide customer service.
- **The requirement to create an organisational social media policy**. Once it has been written, all of the staff who will use social media need to be made aware of it. It should include procedures for staff who will respond to received messages and information requests.
- **An outline project plan showing timescales and tasks.**
- **Targets you will set and how you will monitor and evaluate the use of social media.**
- **Outline the type of content you will post.** You could post about new stock coming in and, for example, articles about new innovations and techniques in the building world.

You should discuss your plan with some of the staff who will implement it to see if they think it is feasible. Once you have a plan, you need to present it to the owner of the organisation for review. Your presentation will need to be professional and focus on the organisational benefits of what you propose. You are likely to need to make some refinements based on the feedback you get from the owner of the organisation.

Getting ready for assessment

Anita is working towards a BTEC National in Computing. For learning aim A, she was given an assignment which asked her to create a presentation aimed at local business people about how they can use social media in their organisations. Anita shares her experiences below.

How I got started

First, I created an overall structure for the presentation by listing the main things I needed to cover. These were:

- an outline of the different social media websites and their audiences
- the different ways in which organisations can use social media, along with some examples
- the risks and issues for organisations of using social media
- an evaluation of the organisation's business use of social media.

I collected all my class notes on this topic into a folder and divided them up into the sections listed above. I then chose a PowerPoint® slide template and created title slides for each section. I also did some research to find examples of organisations that use social media for different purposes, took screenshots of these and copied them into my presentation. I found this approach worked well as it gave me a structure around which to build the presentation.

How I brought it all together

I worked through my notes for each section and used a highlighter pen to pick out the main points. I used these as the bullet points on each slide of my presentation, making sure I didn't have more than about five or six small bullet points on each slide, otherwise the text would have become too small to read. I also checked that I just had the main points on the slides and I rewrote the rest of the text from my notes into the slide notes section, rewording them as if I was talking to an audience. I also added some further screenshots to illustrate the text.

In the section about target audiences and site profiles, I added some links to the external websites that I had used in my research. Once I had completed the presentation, I read it through as if I was actually giving it to an audience and I adjusted some of the slides to make them clearer. I added more information to the slide notes where I felt it was needed.

What I learned from the experience.

I found that the notes I made in class were fine on some topics but too brief on others. I wish I had been more consistent in my note-taking as I had to spend quite a lot of time doing research to find out about some topics which had been covered in class but where my notes were not adequate. I found the last section, the evaluation, hard to write. My first attempt was really just an explanation of the risks and issues of using social media. I had to take each point and think hard about how it might relate to a small organisation and their business use of social media. In the end, I decided that this was hard to cover in the presentation slides themselves and that I needed to write most of the evaluation in the slide notes, where I weighed up the advantages and disadvantages of different aspects of social media use, relating these issues to a small business.

Think about it

- Are you taking class notes and collecting the handouts your teacher has given you? These will be really helpful when you come to write your assignments, so keep them safe and organised in a folder.
- There is a lot of information about social media on the internet. You can use this in your assignments but check that it is up to date as this is a rapidly changing sector. You can use direct quotes only if you clearly reference them; otherwise you will need to rewrite the information that you find in your own words.
- Making a plan for completing your assignments is important. You must hand your assignment in on time, so creating a plan with timings can help you make sure that you have everything ready by the deadline.

The Impact of Computing 9

Getting to know your unit

Assessment
You will be internally assessed by assignments set by your centre.

This unit will help you to understand the impact that computing developments by an organisation have on both the organisation and wider society. The social impact includes human interactions, employment, environmental and ethical issues. During the course of this unit, you will develop a plan to implement a computing technology development in an organisation and then review your plan using feedback from others to analyse the skills, knowledge and behaviours that you practised during this activity.

How you will be assessed

This unit will be assessed by means of a series of tasks that will be set and marked by your tutor. Throughout this unit, you will find assessment practice activities that will help you work towards your assessment. Completing these activities will not mean that you have achieved a particular grade, but you will have carried out useful research or preparation that will be relevant when it comes to your final assignment.

In order to fulfil the tasks in your assignment, it is important to check that you have met all of the Pass grading criteria. You can do this as you work your way through the assignment.

If you are hoping to gain a Merit or Distinction, you should also make sure that you present the information in your assignment in the style that is required by the relevant assessment criteria. For example, Merit criteria require you to analyse and discuss, and Distinction criteria require you to evaluate.

The assignment set by your tutor will consist of a number of tasks designed to meet the criteria in the table.

Assessment criteria

This table shows what you must do in order to achieve a **Pass**, **Merit** or **Distinction** grade, and where you can find activities to help you.

Pass	Merit	Distinction
Learning aim A Understand the impact of developments in computing on an organisation		
A.P1 Explain the impact that developments in computing have had on an organisation. **Assessment practice 9.1** **A.P2** Explain the likely impact of an emerging technology on organisations. **Assessment practice 9.1**	**A.M1** Analyse the risks related to implementing a new computer system in an organisation. **Assessment practice 9.1**	**A.D1** Evaluate the impact that implementing a new computer system can have on an organisation. **Assessment practice 9.1**
Learning aim B Investigate the impact of developments in computing technology		
B.P3 Assess the potential ethical and environmental impacts of developments in technology. **Assessment practice 9.1** **B.P4** Explain how lack of understanding or access to IT can disadvantage certain groups of people. **Assessment practice 9.1**	**B.M2** Analyse the benefits and disadvantages of the social impact of computing technology developments. **Assessment practice 9.1**	**B.D2** Evaluate the impact that the implementation of a specific development in computing technology has had on wider society. **Assessment practice 9.1**
Learning aim C Develop a plan to implement a computing technology development in an organisation		
C.P5 Produce information from a variety of stakeholders to explain the potential impact of a suggested computing technology development implementation on an identified organisation. **Assessment practice 9.2** **C.P6** Develop a plan to implement a technology development within an identified organisation and to manage the associated risks. **Assessment practice 9.2**	**C.M3** Analyse the scope, boundaries and constraints of a computing technology development implementation plan for an identified organisation. **Assessment practice 9.2**	**CD.D3** Use feedback to evaluate the plan to implement a computing technology development in an identified organisation and the suggested improvements. **Assessment practice 9.2** **CD.D4** Demonstrate individual responsibility and effective self-management in the development and review of a plan to implement a computing technology development. **Assessment practice 9.2**
Learning aim D Review a plan to implement a computing technology development in an organisation		
D.P7 Review a plan to implement a computing technology development in an organisation, considering feedback from others and identifying possible improvements. **Assessment practice 9.2** **D.P8** Review the potential social impacts of a plan to implement a computing technology development in an organisation. **Assessment practice 9.2**	**D.M4** Justify the choices made to manage the risks associated with a computing technology development implementation within an organisation. **Assessment practice 9.2**	

Getting started

The mobile phones available 15 years ago were very different from the Android, IOS and Windows® operating system mobiles that many people have today. What differences have these modern phones made to our lifestyles?

A Understand the impact of developments in computing on an organisation

Computing hardware and software continue to develop at a rapid pace and IT professionals in organisations need to keep up to date with these developments, as they provide the organisation with new opportunities and new ways to do business.

Hardware and software developments

Both hardware and software continue to evolve rapidly as new products are brought to market. Advances in technology continue to deliver faster and more sophisticated products.

Increased power of hardware

Hardware has developed significantly in the last few years, and increases in processing speeds and RAM capabilities have allowed systems to be more productive with less strain on the hardware. The fastest consumer processor (2016) is running at 4.7 GHz – this is nearly five times the speed needed for running the most basic of tasks (browsing and word processing).

Faster processors are needed when working on high definition tasks, such as 3D graphics, video editing and special effects. Online gamers require a fast processor with powerful graphics and a fast internet connection. Power users will want to have many apps open at once, often working with large documents, so need high processing speeds, along with a large RAM, to perform at optimum levels. Organisations require servers which can support many hundreds of concurrent users, requiring multiple computers with powerful multi-core CPUs connected to multi-terrabyte disc farms.

Sophistication of software

With the increasing power of hardware software applications have been developed that can carry out increasingly complex tasks.

With more functionality being added to software to reduce the workload on organisations, staff need to be trained on a regular basis to be able to perform all the possible tasks. More sophisticated software packages are capable of performing complex **algorithms** multiple times in the background, while also performing the day-to-day tasks. This can lead to a higher productivity level within the organisation, but this comes with a cost. The more advanced the software, the higher the purchase price, which comes with the additional cost of training the staff to use the new developments. In the future it is likely that the increasing sophistication of hardware and software will allow computers to take over some of the tasks humans currently do, such as driving cars, cleaning the house and carrying out surgical operations. Based on the past experience of how computers have changed the way we do many things, it is likely that computers will continue to change how we work and spend our leisure time, and many other aspects of our lives.

Key term

Algorithm – the solution to a problem. Program designers use algorithms to explain how parts of a program work.

Research

Robotics is one area where increasing sophistication of hardware and software has an impact. Research what robots can currently do and what they are likely to be able to do in the near future.

Mobile computing

These are hardware devices that we can carry with us, for example mobile phones, tablets and laptops.

Mobile applications

As the capabilities of software and web applications have increased, mobile applications have developed alongside them. Mobile phones have become more powerful over

the last few years and this has led to a switch from desktop computers. Applications on a mobile phone are becoming increasingly popular and email clients can be loaded onto a phone in no time, which enables the user to send and receive emails at any point during the day. Microsoft® have now introduced the Office 365® package to the mobile market, encouraging users to create documents on the move. The term 'working day' is now becoming outdated due to the development of the mobile market.

With the developments in software and the newer languages being developed (for example, Ruby and Python®), organisations are able to create bespoke software for their specific requirements. This can be loaded onto a mobile platform, which improves productivity for the workforce as it enables them to work 'on the move'. However mobile platforms provide some challenges for the software developer. Smaller screens and the requirement to type on a small on-screen keyboard mean that applications have to be designed carefully, with versions of desktop software adapted specifically for mobile devices.

Mobile hardware

The demand for the mobile phone and tablet PC has grown over the last decade. Ofcom state that 66 per cent of adults owned a smartphone in the UK in 2015 compared with 61 per cent in 2014. The increase in numbers has created a requirement for more powerful phones with more functionality. For example, a current leading handset comes with a 5.5 inch screen, 4GB of RAM and 32GB of storage (see Figure 9.1).

Figure 9.1: Samsung® Galaxy S7 Edge

With hardware for mobile phones getting smaller, companies are able to add more to each component, to create a personal computer that you can put in your pocket.

The early smartphone processors were not capable of performing many tasks at once. The latest processors, for example Snapdragon 625 (see Figure 9.2), bring increased battery life and this model uses 35 per cent less power than previous versions, giving an extra 21 minutes usage each hour. This enables an increase in the functionality that the phone can accommodate.

The Samsung® S7 holds 4GB of RAM, which is comparable with most budget desktops. While this is needed to cope with the increased functionality, the RAM will also help a user to have more than one application open at once, enabling seamless moves between email and other applications.

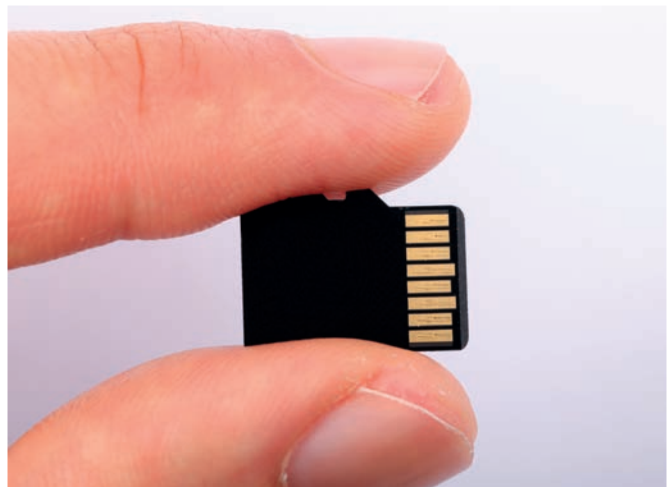


Figure 9.2: Micro SD card

SMS text messages were the main storage element on the first generation of mobile phones. An SMS text message containing 16 characters requires 1KB of storage and today's phones need to be able to store locally at least 1000 messages in a thread (conversation) as well as data about yourself and applications. Therefore storage space had to grow. The iPhone 6s (2015 model) has 128GB of internal storage without a micro SD extension. 128GB is sufficient to hold 56,998 images, 18,639 songs, 5,699 applications, 2,185 games, 85 SD videos or 32 HD videos.

The latest Android® and Windows® operating system phones are able to exceed this capacity, and if this is extended with a micro SD slot (see Figure 9.2), mobile phones have the capability to hold the same amount of data as a small laptop (e.g. 500 GB).

Research

What are the specifications in terms of RAM, processor speed, battery life and storage space for the latest smart phones available as you read this? How do these changes impact on how these devices can be used?

Improving the competitive edge

Developments in computing can have positive and negative effects for organisations. With these developments come greater opportunities for increased business and increased capital (profit). Organisations in many industries are using developments in computing to gain a competitive edge. Many developments also provide increased convenience for users; but there are negative impact too. For example the development of contactless payment has brought with it the benefit of not having to remember a PIN (personal identification number), but has the negative effect of making theft easier.

Organisations are looking at ways in which they can outsmart their competitors and generate more business before rivals come up with competing developments. In manufacturing the use of computer controlled robots allows production lines to be more efficient, lowering costs. Robots are able to assemble items more consistently and accurately than human workers, which means high quality products. Producing high quality items at a lower costs helps give companies a competitive edge. On-line stores are another example of how computers provide a competitive edge. Unlike traditional shops these are available 24/7 and run with far fewer staff.

Upgrading hardware

Many organisations regularly upgrade their hardware to take advantage of more sophisticated software and improved energy management.

The development of software, for example more sophisticated and powerful packages, has created a need for regular updating of hardware. As software is getting more powerful it will use more processing power and RAM; it may also perform better with multi core CPUs. For example, a video editing app such as Adobe After Effects will use 74 per cent of 3 GB of RAM to render a small video. This will then create a slower computer system for the duration of the rendering process.

To accommodate these developments, regular maintenance needs to be carried out and hardware may need to be replaced so that the computer will be able to complete the software's instructions while allowing the user to carry on with day-to-day tasks. In some cases hardware replacement may be easier and less time consuming than upgrading, and due to the time and effort required by technicians to complete an upgrade, it may be cheaper too. However disposing of the old hardware may be problematic and create environmental issues.

Power efficiency is a vital factor in ensuring that a computer will run economically. A normal office space might consist of between ten and twelve computers, which can generate a significant amount of heat. This can mount up and create a very hot environment.

Developments of '**chipsets**' can help to reduce the power usage of some desktop machines and laptops. The Haswell chipset, developed in 2014 for Intel CPU based systems, was designed to reduce the heat resonating from the base of a laptop. This was an added safety feature, as many laptop users rest a laptop on their upper legs. The chipset was prebuilt into laptops sold after 2014 and helps to reduce heat generated around the laptop's CPU.

Key term

Chipset – this is the collection of circuits on a component such as a motherboard or a graphics card. Each release of a chipset is a new design that usually offers improved performance, often with faster operation and a lower power requirement.

Cloud and hybrid cloud technologies

Since 2000, the rapid development of cloud storage and **software as a service (SaaS)** has brought about a new digital age where you have the potential to access work and personal documents from anywhere in the world with an internet connection.

Developments of web 2.0 technologies have stimulated the use of cloud computing. One example is the Google Docs™ web-based word-processing program which enables several users to work collaboratively on the same document simultaneously.

However, there are concerns about security for businesses using the services that cloud computing offers. With cloud data held externally to an organization, often under the control of a third party, and available over the internet, keeping the data secure becomes more complex. Cloud technology has developed alongside mobile computing because users want more flexible methods to store data, which provide version control when accessed from multiple devices. Cloud computing is able to offer a wide range of services that can bring businesses together and increase productivity.

Hybrid cloud computing has developed as an alternative cost-cutting option for businesses. As the cost of having an 'on-site' networked system increases and the cost of cloud networks decreases, businesses are using a mixture of data storage on their premises and on private or public cloud networks. By being flexible about where it houses the majority of its network, a business can be more cost efficient as prices change. Because of the combination of

data storage options, hybrid cloud can be more complex to set up; but it does provide the option of continuing to store the most sensitive data in-house, while less sensitive data which needs to be widely accessible can be stored in a private or public cloud.

Key term

Software as a service (SaaS) – software as a service lets you use apps that are licensed by a remote provider through cloud technology.

Changing markets and new opportunities

The ever-changing world of computing has an inevitable impact on industries such as banking, teaching and retail, affecting small companies as well as multinationals.

Decline and development

As many traditional retail outlets in the high street are shutting up shop, many online retailers find that their marketplace is booming while others find that their way forward is to combine these selling platforms.

Brick and click

There has been a decline in shops 'on the high street', and businesses that would occupy stores in most towns and cities are moving online and towards a 'click' marketplace. Major retailers such as Next and Tesco plc will still have a presence on the high street, but they have also developed an online presence to give the customer a complete shopping experience at home. Customers are able to log on to their websites, do a weekly shop and have it delivered, but they are still able to go to a store and walk round to select their products. This has produced a growth in their sales and market ownership. Small high street shops have struggled to compete with this model.

Independent stores

Small businesses that serve the local community and which may consist of only one store have been affected greatly by the rise in online shopping. More and more shops (11.7 per cent on average in 2015) are being vacated as businesses are unable to keep up with the growing online trend, leaving many high streets in British towns with empty shops. However it has led to a growth in the delivery business, required to home-deliver items purchased on-line. Many independent niche retailers have now moved off the high street and set up on-line using social media site to promote their products and sites such as eBay and Amazon Marketplace to sell their goods.

One advantage of setting up a small business in this way is that it has much lower start up and running costs than a traditional shop: the market is now open to a much wider range of entrepreneurs than was previously possible.

Woolworths Group plc is an example of a solely 'brick' environment that went into administration due to the rapid development of online sales.

Discussion

Research some retailers that have disappeared from the high street in recent times, such as BHS, Blockbuster Videos Inc, Comet and Virgin Megastore. Why do you think these companies encountered challenges as market conditions changed?

Online only

Online stores such as the grocery store Ocado Group are growing each year. With more and more businesses moving online and creating a crowded market, online-only stores are needing to come up with new ways of generating business. Amazon and eBay have been at the forefront of these developments. Amazon and eBay both have mobile applications which some vendors pre-install into Android and iOS operating systems.

Mobile market places are developing into larger arenas with more organisations creating mobile applications for shopping on the move.

New markets and opportunities

The conventional business would normally specialise, for example Tesco plc is a supermarket. Recent developments have forced businesses such as these to develop new ways of keeping and growing their customer base.

Amazon has been a huge stalwart in the online trading platforms for over a decade, but even a company of this size needs to have new ways of improving its customer numbers and increasing profits. The development of Amazon Prime has created a new strand to Amazon's plan, enabling them to offer a next day delivery service for most products, discounts on selected items and access to an online streaming service which rivals Netflix. While this latter service is still in its infancy (Netflix has a bigger customer base), Amazon is able to develop the service to add more functionality for the customer, so giving them the advantage of offering more for less. Also, by introducing the Kindle e-reader Amazon was able to build on its original book-buying customer base while introducing new technology. The Internet opens up new markets, even for small companies.

Traditional methods of selling goods and services (such as having a high street shop) tend to be geographically limited but anyone selling via a web site or other on-line methods such as eBay or Facebook® has the potential to attract customers from across the globe.

By developing and changing as technology advances, an organisation is able to keep up to date with what the customers want and this gives them an advantage over their competitors. This will keep the client base at a stable level and will help to keep profit levels up. However, these developments can come at an increased cost, as the newest technology is likely to be more expensive than existing technology.

Research

In 2016 Amazon started UK testing of its plans to deliver some packages using unmanned drones. Research how the testing is progressing. What are the advantages and disadvantages of delivering packages by drone?

Balancing growth against decline

It is always difficult to make sure that everything is balanced but still continues to grow in new areas. The development of 'click-only' services has had an impact on the high street. Larger companies are trying to find ways of keeping a presence on the high street and still develop online. As services move towards automation, this will have an impact on the number of jobs that can be offered in the future and many employees may find themselves made redundant. The types of jobs that are in demand have also changed. Manual work such such as assembly work in manufacturing companies have declined but technology related jobs such as programming and engineering are growing. Developments in computing have changed the way we do many things (including taking photos, listening to music, banking, shopping and many more). This has led to decline in some industries but has largely been balanced by growth in new industries that develop or make use of the new technologies.

Analogue music and photography still have their place with purists who believe that they provide the most accurate reproduction of sound and vision. For many, the digital equivalents are preferable because of the small physical size of players, their portability and the ease of downloading or purchasing items online.

Cost reductions with improved customer service

By making use of the development of automation in technology, organisations are able to increase their productivity. Customer service is an area that has undergone one of the biggest changes owing to developments in technology. With the use of improved **frequently asked questions (FAQs)** on websites, customers are able to either find their answers there or speak to an advisor, using **internet relay chat (IRC)**, who will know what the customer is looking for or respond to their queries.

Automation also allows organisations to increase productivity across the network while reducing costs as the cost of employing personnel to carry out a task will be more than it would be for an automated system to carry out the same task. This system will also be able to complete the task in a fraction of the time (depending on the task) that it would take a person.

Key terms

Frequently asked questions (FAQ) – a common feature of websites and other documentation. These are simply a list of common queries that people make along with their solutions.

Internet relay chat (IRC) – this is the technology behind instant messaging that enables you to write messages and receive them from your friends or colleagues while using a computer.

Big data opportunities and challenges

The advancement of technology has resulted in more data being produced at a rapid rate. This comes with both opportunities and challenges. One of the opportunities is the ability to analyse the data quickly, and very large data sets can give a lot of information to an organisation, for example in the form of their customers' favourite products or their search history on a website. Companies such as eBay use this to create a personalised homepage for a customer, determined by their search history.

With all this data come challenges that can affect the way in which an organisation operates. It is necessary for an organisation to be able to respond quickly to the data it collects, and using visualisation (page 417) helps to speed up the analysis of it.

Understanding the data that is analysed can be another challenge as, without meaning, this data is nothing to the

organisation. Applying context to the data that they collect will help an organisation to m ake informed decisions and act quickly to address any shortfalls.

Emerging technologies

There are always new advancements in computing being brought to the marketplace.

Internet of things

The internet of things (IoT) is technology that allows you to connect devices that would not normally be considered to be computers (such as fridges, heating systems and washing machines) together over the internet. This technology is becoming more and more popular in modern society. Some of the most popular are home control systems, such as Hive from British Gas. This technology allows you to interact with devices in your home by means of your mobile phone.

IoT has endless opportunities for businesses. For example, farmers can use this technology to monitor the weather and automatically trigger watering systems when it has been dry for several days. In office-based businesses, IoT can even be used to boil the kettle when the sensors determine that the employees have dropped their productivity rate.

With all these technological advancements, there are inevitably going to be challenges that will need to be faced. Online security will always be an issue and a key question for any organisation or individual to ask when storing or transmitting data is: Will it be safe? A large amount of personal data can be collected about an individual even in their own home, for example electricity usage or their online shopping habits. This would then be stored on an application server. Would this information be secure?

Research

The Hive thermostat from British Gas allows you to adjust your home heating system remotely using your smart phone. Hive has recently added a number of other home automation devices which can be controlled from your phone. Research what these home automation devices can do. What are the advantages and disadvantages of home automation?

Increasing integration

With the increasing involvement of technology in almost everything you do, you are no longer restricted to using your networks only at home or in the office but are now able to access these from anywhere in the world. Integration between different applications also allows new possibilities. For example the GPS system in your phone could identify when you leave work and instruct your home automation system to turn up the heating and switch the lights on as you approach.

Increasing automation

Automation is an important and fast-moving area of computing in today's world, and the range of its capabilities is expanding. The use of robotics and exoskeletons has led to many remarkable developments. Industrial robots can carry out repetitive tasks like paint spraying more accurately and consistently than human operators and can work 24 hours, 7 days a week. However industrial robots are very expensive and need to be programmed for a specific task and so are best suited to very large volume tasks. There are some types of task which robots are not well suited to, such as intricate assembly tasks or tasks where there is some variation in what has to be done.

Robotics

The field of robotics has developed rapidly over the last decade, and scientists in Japan are close to creating full artificial intelligence (AI). Self-driven cars are being developed by Google™ and Apple® and drones are being tested by Amazon to deliver small packages. These use an in-built computer system to complete the tasks given to them.

Robotics are frequently used in the rehabilitation of amputees, by using state-of-the-art designs to create limbs, for example arms, that are fully functioning. These use the nervous system to receive messages from the brain, which are interpreted into a command or task and carried out.

Exoskeletons

Exoskeletons were originally developed for the military. They are suits that are worn by the user (think *Iron Man* from the comics and cinema screen) which have a computer system built in and which can respond to commands from the user through movement and speech. Exoskeletons offer increased strength and durability, by providing a hard outer casing.

The European Union is sponsoring companies to develop exoskeletons that can be used by people with physical disabilities, which would, for example, have a huge positive effect on people suffering from paralysis, as it could enable them to walk.

Current developments

Wearable technology has been around for a few years, but recent developments by Apple® and Samsung® are starting to dominate the market for wearable devices. The Apple Watch® and Samsung Gear® are developments in this area. The early smart watches were linked by Bluetooth to your phones while they were in your pocket, but they only had limited functionality. Fast forward a few years and you are likely to be able to use your smart watch on its own as a mobile phone.

The use of the fitness range of wearable technology has also increased due to the current trend for being fit and healthy. These watches can keep track of your step count, calorie count, heart rate, etc. Linking to your phone enables you to monitor your exercise regime with a greater degree of accuracy, while interaction with other users provides an element of competition and incentive.

Big data, data warehousing and data mining

Big data refers to the many modern sources of data, in particular from mobile phones and internet use. Data warehousing refers to the technologies used to hold such massive amounts of detail while data mining means the techniques used to gain information from this large mass.

Uses of large amounts of data

Big data is defined as data sets that are too large for normal processing applications to be able to analyse. Data can be captured about anything or anyone at any point during a day and this has to be kept somewhere.

Data can be captured from many sources, such as financial transactions, social media or search engines. This helps organisations to analyse how they can be more effective in the marketplace and how to develop to meet customers' needs. Research and forecasting can also benefit from big data A analysis. Examples include weather forecasting and medical research into factors influencing disease.

'Big data' is not new terminology, but Doug Laney, META group analyst, refined the concept by suggesting in 2001 that large amounts of data can be split into three main categories: volume, velocity and variety.

Big data features

Big data has these characteristics and features:

- volume
- velocity
- variety
- storage
- processing.

Volume

Volume refers to the quantity of data that is generated from a variety of sources. There is normally a lot of data generated at any one time and storing it can be an issue for organisations.

Velocity

This is the speed of generation of data. Data is streaming into organisations at an alarming rate, and often needs to be analysed almost instantly. This can cause a strain on the internal computer systems.

Variety

This describes the mixture of data to be processed. Data comes in a variety of forms, such as simple text-based information, financial information or video and audio. The receiving systems will need to interpret all of this together for it to be of use to the organisation. Sophisticated software is required to enable instant analysis.

Storage

Storing all of this data can be an issue, but developments in software, for example Apache™ Hadoop®, have made this task much easier. Data can be split up and stored on multiple areas of a network, and then it can be linked back together to be analysed when required.

Processing

It will take a lot of processing power to analyse these large amounts of data. The use of parallel processing has enabled this to be done much more quickly than in the past. Parallel processing works by splitting the task up between multiple processors in a system, so enabling a task to be performed much faster than on a single processing unit.

Data warehouse features

The key features of a data warehouse are:

- subject orientation
- data denormalisation
- non-volatility
- large amounts of historical data
- use of queries to retrieve large amounts of data
- both planned and ad hoc queries are common
- controlling of data loads.

Subject orientation

The data that has been collected will be in a jumble with no discernible pattern. Subject orientation will store the data by a defined topic or theme so that it becomes more streamlined and easier to analyse.

Data denormalisation

Data denormalisation allows for simplification and improves performance. Denormalisation is the process of grouping data or adding redundant data to improve performance if a query is run, and is very popular in database solutions. Grouping data together in a table can help to speed up analysis and improve the performance of the system as the search will be carried out in one defined area and not through the entire data set. For example a database might have separate tables for customers and products. The data in the two tables can be brought together into one table so that analysis can show which customers bought which products. With very large datasets it will be quicker to read the data from a single table than to extract it from two linked tables.

Non-volatility

Data storage is an important area. Organisations will want to know that the data they have will not be affected by issues such as the power being suddenly turned off. Having a storage medium that is not volatile will help to ensure that the data is stored correctly and safely and will not be lost if a problem does arise.

Historical data

With the development of digital technology, although the need for keeping historical data has increased, less space is required to store it. Organisations must, by law, keep some data for a certain period of time, which would require a lot of space if it were kept in paper files. Historical data can be very useful to organisations as its analysis may show trends, such as how the weather is changing over time or how people's buying habits are changing.

Use of queries

Queries are a very common way of sorting and analysing data. A planned query will involve applying a series of given steps to the data for the purpose of finding the best data for a given task.

Ad hoc queries will be generated when a need presents itself. These queries will search for certain parameters given by the user and will return the best results found.

Both query types are common when analysing data – planned queries will be used when wanting to look for large similarities within the data, and are regularly used. Ad hoc queries are used when there is a requirement to find detailed aspects of the data from the results of larger queries.

Controlling data load

When analysing the data set, organisations will need to be careful about what is being asked for when loading a dataset. An example of this would be a high security company loading the images of each employee when searching for a date of birth. This is an inefficient way of using the data. Being able to control what is loaded will be an efficient way of reducing the processing power and allow for greater improvements in performance. Historical data can be very useful to organisations as its analysis may show trends, such as how the weather is changing over time or how people's buying habits are changing.

Data mining

Data mining is the process of analysing data, which may be contained in a data warehouse, and attempting to discover meaningful information from it that can be used by the organisation to predict future outcomes, discover new opportunities and improve the performance of the business.

Data mining techniques automatically search large amounts of data to discover patterns and trends that may not be obvious with simple analysis. There are many different sophisticated mathematical techniques that can be used to analyse the data. Some of the most common ones are listed below.

- Cluster analysis. This involves attempting to divide the data into groups or clusters. For example, a supermarket may divide its customers into groups according to whether they are interested in a particular product or based on their previous purchasing history. This information can then be used to target advertising at particular groups.
- Anomaly detection. This type of analysis looks for data that is outside the norm (e.g. values which are significantly different from the usual range of values). Typical applications include credit card fraud detection, where transactions are flagged up as possibly fraudulent because they involve far larger amounts of money or take place in a different location from where the account holder normally shops.
- Association rule mining. This attempts to identify dependencies in the data. The technique was originally developed to allow retailers to analyse customer shopping baskets and identify items that were commonly bought together. So, for example, the analysis might identify that when customers purchase one item they very commonly also purchase another, otherwise unrelated, item. This data can be useful for deciding how to adjust store layouts (which items to place close to each other), for choosing promotions and optimising catalogue design. The technique can also be used for other applications such as medical diagnosis and weather forecasting.
- Data visualisation. This involves presenting the data in charts or graphs to allow users to understand the data more easily and identify patterns that may exist within it.

Although data mining is primarily used by retail companies to identify customer preferences and to use this information to inform decisions about pricing, promotions and product placement, it is also used in a wide range of other applications such as banking (for example to assign a credit score to a customer when deciding whether to lend them money or not), scientific research and network intrusion detection.

There are many data mining tools available, comprising both open source and proprietary software. Examples of well-known proprietary software include IBM SPSS modeller, SAS Enterprise Miner™ software and Microsoft® SQL Server Analysis Service.

Research

All of the well-known proprietary data mining software products have detailed information online about how they can be used. Use an internet search engine to find out more about these products and their capabilities.

Issues and risks

You need to be able to recognise the many issues associated with the way in which computer systems are used and the risks these may present to an organisation.

Security considerations

Security considerations reflect a growing need to be aware of the dangers inherent in an extensive reliance on computing for every aspect of life.

Security will always be a major issue associated with computing and hackers are becoming more adept at gaining access to computer systems. All organisations need to be aware of security and establish how they are going to ensure that any personal or sensitive data is kept secure.

As organisations become more cloud-based and rely heavily on this technology, security needs to be a major factor in any decisions made. How secure will the network be? How might someone try to get into the network? These are questions that will need to be answered robustly to ensure the safety of the organisational infrastructure and the privacy of the data held there.

Criminals and terrorists

Recent times have required organisations to be even more vigilant with the security of their data to respond to the threats posed by criminals and terrorists.

Data theft or destruction

There are always going to be hackers trying to get into major organisational network infrastructures to take advantage of vulnerabilities and try to steal secure data. Many have succeeded in this, for example Snapchat, the Playstation® networks and Xbox® have all been affected by theft of data.

In larger organisations or websites, a spike in online traffic can be seen by criminals, which tempts them to try to get into the servers to see if there is anything that can be used for financial gain or some other illegal purpose.

Once inside a system, data theft occurs when information, such as customer names and bank details, is obtained by the hacker who can then use or sell this private data.

Destruction of data is when the hacker deletes or corrupts information in a system because they wish to harm the organisation.

PAUSE POINT Why would a hacker want to delete or corrupt data in a computer system?

Hint What reasons would there be to damage an organisation??

Extend What organisations do you think would be targets for such an attack? Why?

Fraud

Fraud is an inherent problem with modern computing as we all rely upon the internet so much and there is very little that you can do to confirm the identity of someone at the other end of a communication. Fraud is also a very low risk activity for a criminal, for example sending out a phishing email is unlikely to result in a criminal prosecution for the sender especially if they are in a different country.

Denial of service

Denial of service attacks are carried out by sending a web site so many bogus requests that it is overloaded and cannot respond to legitimate users. Many networks are able to deter these attacks with robust security, but these attacks are becoming far more complex and networks need to constantly change in order to continue to prevent them.

Key term

Denial of service attack – is a coordinated attack on a website by many computers simultaneously. The aim of the attack is to prevent the website from being able to function by overwhelming it with traffic.

Blackmail

Blackmail is another opportunity for a criminal who is able to obtain private information about someone that they do not want to be made public. Cyber criminals may also use a number of methods to blackmail an organisation. For example they may threaten to launch a denial of service attack on a company's e-commerce web site unless they are paid money.

Maintaining compatibility

As systems are developed, there is a need for them to be compatible with both newer and older versions of software.

Internal systems

This can be difficult as some **bespoke** software may not be compatible with a newer system as it would have been written for an older one. Moreover, some older software will need to be redesigned in order to work properly. For example, Aptos is purchase order software used in the education system, but was not compatible with Windows® operating system 8 when the latter was released. Not only can updating old software be costly and time consuming, it may be difficult unless the software is fully documented.

Key term

Bespoke – is software that is specially written for a purpose, as opposed to off-the-shelf software which can be purchased from a shop or downloaded.

Research

Cyber criminals are always looking for new ways to attack systems. Research the most recent security incidents and find out what happened, what the impact was and how systems should be protected in the future against such attacks.

External systems

In order to get ahead of their competitors, an organisation may upgrade their internal systems to the latest software.

This may cause problems, as some of their external collaborators will not have the latest updates installed. People working from home may need to update their software or hardware and this is likely to be at a cost.

For example, if a company updates its website to take advantage of some of the latest features of web browsers, it may fail to work correctly or even at all for users who have older versions of web browsers installed on their computers.

Increasing complexity of IT systems

Key term

Mission-critical – a mission-critical system is one which, if it failed or suffered from errors, could cause serious problems such as disruption affecting a large number of people, significant financial loss or even loss of life. Examples of mission-critical systems include air traffic control, vehicle control systems and medical systems.

IT systems control many aspects of our lives and have many connections with other systems, all of which are becoming increasing complex. For example, IT systems control banking and financial systems, electricity generation, supply and distribution and transportation systems such as trains and trams. Some of these systems control **mission-critical** applications where failure will cause disruption or even loss of life.

Transportation systems have used computer-controlled signal lights for decades, but now trials are taking place for cars and trucks that can be driven by a computer, replacing the human driver.

Military systems have many uses for computer control. Aircraft such as the Eurofighter Typhoon are unable to fly without the help of their computer control systems. These mission critical systems have many advantages over human or mechanically controlled systems: they are more accurate, more reliable and can perform tasks more efficiently.

Reliability

There are real difficulties associated with ensuring that complex systems are reliable, fully tested and 'failsafe'. The more complex a system, the more detailed the testing should be, but there is also a greater possibility that a complex system may have undetected errors which may cause it to fail or respond incorrectly in circumstances that have not been envisaged.

We have now reached the point where we are creating very complex systems that are able to respond to real-world situations, such as driving a vehicle through a city, which are impossible to fully test. For example, there are so many unexpected situations possible on the city roads that cannot be anticipated.

Case study

Tesla Model S

At the time of writing, the Tesla Model S is probably the most sophisticated car available as it is almost able to drive itself. It has an autopilot facility but it is not a fully autonomous system as the attention of the driver is still required. It uses computer-controlled video cameras, forward-looking radar and ultrasound detectors to provide adaptive cruise control (which adjusts the speed of the car automatically based on the traffic ahead of the car) and collision avoidance (which applies the brakes if it detects an obstacle in front of the car). However, the system is only suitable for motorway driving and cannot be used for urban driving as it does not detect pedestrians or cyclists and does not respond to traffic lights.

Collision avoidance systems should make cars safer as they can apply the brakes when an unexpected obstacle appears even if the driver is distracted. However, on 7 May 2016 the first fatal road accident occurred involving a Tesla Model S in autopilot mode. The car collided with a truck which had pulled out in front of it. It is thought that the autopilot system failed to recognise the white painted trailer of the truck against the bright sky. The accident highlighted the problems that self-drive cars face and the difficulty of testing every possible circumstance. There may also have been an element of driver error in the accident as the driver is supposed to remain attentive while using the autopilot system. This in itself raises a question of whether we are too willing to trust systems on which our life may depend.

Developing mission-critical systems is, in itself, a complex, expensive and time consuming process. Typically the development of such systems will involve:

- redundant hardware so that the system can continue even if there is a hardware failure
- fault monitoring so that the system can detect faults such as failed sensors
- a rigorous change-monitoring process, with full retesting if any changes are made
- software developed to a formal standard, such as IEC 62304, which is the standard for medical device software
- use of formal code reviews during the software development phase (a code review is when someone other than the programmer who wrote the code systematically inspects it to attempt to identify any errors and weaknesses)
- rigorous testing which will often take more time to carry out than the code itself takes to write.

Despite all these precautions, there can never be an absolute guarantee that any software system will always perform correctly in complex situations.

PAUSE POINT Prototypes of computer-driven vehicles are being tested by various manufacturers. What testing should be used to guarantee these vehicles are safe?

Hint Research a current computer-driven vehicle. When was it first produced? How is it being tested? When is it expected to be available to the public?

Extend What do you think should be the most essential tests to ensure that a computer-driven vehicle is safe to use on public roads?

Changes in working practices

Changes in working practices include:

- remote working
- new office practices
- upskilling of workforces
- reduction in low-skilled jobs
- new working styles.

Remote working

Working away from a desk is becoming more and more common. Employees are able to access work emails from mobile phones and work on documents by means of a tablet or laptop whenever and wherever they like, as long as they have an internet connection. The increased availability of broadband internet in most parts of the country and the availability of low-cost high-performance computer equipment and peripherals has encouraged this trend. While there are many advantages to this, it can cause problems with work–life balance. Employees may feel under pressure to check and respond to emails outside work hours.

New office practices

As many people own their own smartphone and/or tablet, they commonly bring these devices to work with them and may feel more comfortable using their own devices to access work emails and documents rather than the systems provided by their employer. This office practice is known as bring your own device (BYOD) and many employers are happy for their employees to do this and will provide **wireless access points (WAPs)** within the office for them to connect to. However, BYOD can raise a number of issues. Support can be a problem because of the very wide range of hardware and operating systems that employees may use on their devices.

IT support departments within a company are used to working with a limited range of hardware and software but, with BYOD, they may have to deal with support issues from many different hardware and software systems. The other major issue is security. Some of the wide range of different devices connecting to the company network may not be protected properly from viruses and other security risks. This can create risks for the company network and the company may need to require that certain rules are applied to BYOD devices, such as installation of anti-virus software. There is also the chance that employees' BYOD devices could be lost or stolen and that confidential documents and emails downloaded onto the devices could fall into the wrong hands. Most companies which allow BYOD will have a policy which states the rules and procedures for employees using their own devices at work.

BYOD is also often used in education, where students can use their smartphones to take part in quizzes and surveys.

Key term

Wireless access point (WAP) – most homes have a wireless access point (WAP) built into the internet router so that laptops and other devices can use WiFi to access the internet. Many organisations also have dedicated WAPs arranged so that WiFi is available to staff and visitors. A dedicated WAP is connected to the network and does not include a router. Companies providing WAP access to their networks for employees' BYOD access use authentication software to ensure that only authorised employees can gain access to the network.

Link

BYOD and its security implications are also covered in *Unit 7: IT System Security and Encryption*, in the section on internal threats.

Upskilling

With IT systems becoming more complex, employees need the skills to be able to use the system effectively. Software applications (and operating systems) change rapidly and new versions add features that not everyone is familiar with. Therefore employees need regular upskilling to get the best out of the applications they use. Microsoft Office, for example, is upgraded to a new version every few years. While training can be costly and time consuming, most employers recognise the benefits in terms of efficiency and staff morale of having a highly trained workforce.

Some organisations may delay upgrading applications and operating systems because of the retraining required, not just of users, but of support staff too. However, many software companies will stop providing support for old versions of software after a few years so that organisations are often forced to upgrade.

Continuous professional development (CPD) within the organisation should be done at regular intervals to upskill the workforce. Whenever a new system is implemented, staff should be trained to use that system, as there will be updates and new features that can help improve the performance and productivity of the workforce.

Key term

Continuous professional development (CPD) – refers to any training or activities that an employee undertakes to keep their knowledge current or to enhance their skills.

Reduction in low-skilled jobs; new working styles

The requirement for low-skilled manual workers has decreased in some industries, especially manufacturing where automated machines can complete repetitive tasks more quickly, accurately and reliably than human workers. On car production lines, for example, robotic machines are used extensively to carry out tasks such as fitting windscreens and doors.

Over the last 50 years, the employment structure in the UK has changed dramatically. Traditionally, large numbers of people were employed in jobs involving manual labour, such as manufacturing, farming and coal mining. For a number of reasons, not all to do with developments in technology, there are now far fewer of these types of jobs in the UK. Many manufacturing jobs have moved to the Far East and China where labour costs are much lower. Most of the manufacturing left in the UK is high technology based, for example aerospace and car-making. The manufacturing jobs that have been lost have, in many cases, been replaced by desk-based jobs which rely heavily on computing, for example insurance, banking and customer service.

All these changes mean that a workforce is required that is well educated and skilled in using information technology. Some low-skilled manual jobs remain but these tend to be in areas such as retail and catering, and tend to be lower paid.

In the future, it is likely that technology will continue to replace lower-skilled jobs. Some lower-skilled jobs will remain but the demand for well-qualified workers with computing and engineering skills is likely to increase.

Increasing reliance on IT

With many organisations relying heavily on computer systems, the consequences for failure can be very significant. Online stores such as Amazon are 100 per cent dependent on their IT systems and, should they fail, the financial consequences would be very severe. Banking and mobile phone service providers are other examples of industries which rely very heavily on IT, and system failures would have serious consequences for the company and have an impact on the general public.

IT system failures can occur for three main reasons.

- **Software failure**. Software crashes and errors can cause problems for businesses. As mentioned earlier, mission-critical software needs to be tested very carefully but it is also important that business software is tested fully to avoid problems during its use. Software failures can cause businesses to lose money as they may be unable to carry out their normal business until the error is corrected. It can also be embarrassing for a company and can frustrate their customers. For example, recently there have been several occasions when banking systems have suffered errors and customers have not had their wages and other credits added to their accounts until the problem has been resolved.
- **Hardware failure**. Hardware such as processors, disc and networking equipment will fail from time to time. Business-critical computing systems will normally use redundant hardware with dual components. Discs, for example, will be configured using **RAID** disc mirroring, with multiple discs used in sets to ensure that if a single disc fails the system can continue without interruption. Server hardware allows failed components such as discs to be 'hot swapped' without closing down the system. Server computer systems also commonly have uninterruptable power supplies using batteries or generators which continue to supply electricity even if the mains supply fails. All these features add to the cost of the computer system but they do prevent hardware failure from taking a company's computing resources off line.

Key term

RAID – stands for redundant array of independent (or inexpensive) discs) and is a technique that uses multiple hard discs to provide higher performance and fault tolerance. There are a variety of types of RAID configurations. The most commonly used RAID 5 uses three or more discs and allows the system to continue uninterrupted if one of the discs fails.

- **Physical disaster**. While software and hardware failures can temporarily disable part of a company's computer system, physical disasters such as a flood, fire, earthquake or even terrorist attack could destroy not just computer server systems but the facility in which they are housed. In companies where the availability of their computing resources is essential, disaster recovery planning is carried out to prepare for such an eventuality. Disaster recovery planning involves preparing an alternative site from where the company's computer systems can be run. There are a number of different options for this.
    - Cold site. This is essentially just an empty computer room in a building that is geographically remote from the business's main computer site. Should disaster strike, the company will need to purchase and set up server computers at the cold site and install, from backups, the software and data required to run the business's systems. Cold sites are often provided by companies that specialise in disaster recovery. Businesses wishing to have the facility available pay the disaster recovery company a fee to be able to use the backup site whenever required. Using a cold site is the cheapest disaster recovery option but there is also a considerable delay in getting the original system up and running (perhaps days or even weeks).
    - Warm site. With a warm site, suitable hardware is installed and ready to run the company systems. To get the system running all that is needed is for the latest backups to be restored on the system. The delay in getting the company systems running again after a disaster is much shorter (maybe in terms of hours) than with a cold site but the cost of maintaining a warm site is much greater.
    - Hot site. This option maintains a complete running duplicate of a business's IT systems. The main system's data is usually synchronised with the hot site using network links. The advantage of this is that a hot site can take over from the main system's site very quickly should the need arise but the drawback is that the cost of maintaining a hot site is very high.

The planning which goes into deciding how to protect a company's IT systems from disaster is known as business continuity planning. The key decision that a company needs to make is based on how long it can afford to be without its computing systems. This is known as the recovery time objective (RTO). This can be based on a number of factors but, for example, an online store which takes perhaps £5,000 of orders per hour will lose £120,000 if its systems are offline for a day (and nearly £1 million if

they are down for a week). The business would also have its reputation to consider as some customers might go elsewhere to purchase items if the store was unavailable, and may not return. Therefore spending money on a disaster recovery plan is likely to be considered a wise business decision.

Link

Disaster recovery is also covered in *Unit 7: IT Systems Security and Encryption*, in the section on IT disaster recovery plans.

In the future, although it is likely that hardware will become more reliable, our dependence on IT-based systems is also likely to increase. Therefore ensuring that these systems are always available, safe and secure from misuse will remain a priority, as the consequences of failure will have a huge effect on our lives.

Information overload

With many organisations, there will be departments that need to use the same data. Having a large quantity of data that is duplicated can require a large amount of storage space and cause undue pressure on the processors that are needed to work with the data.

This can be both expensive and difficult for the organisation. Where data is duplicated, updates become difficult, since all versions of the data must be updated. Sharing data can also be problematic since, if the data needs to be changed, whoever is changing it needs exclusive access. There can also be problems of version control, that is, ensuring that everyone is using the latest version of (for example) a document. An intranet, for example SharePoint, would help here to ensure that any data used can be accessed by anyone in the organisation. This reduces the risk of duplication.

Information overload is an ever-present risk in the modern world, making the difficulty and expense of processing large quantities of data into useful information worthwhile.

B Investigate the impact of developments in computing technology

You need to research and understand how computing technology developments have had an impact on personal communications, employment and the environment, along with the ethical issues arising from these developments.

Social impacts

There have been massive changes in the way people communicate during the last decades, with increasing reliance on social networking for human interaction. For many people, there are fewer opportunities to practise social skills, greater isolation, and age-related issues associated with using technology. The ease with which contacts can be maintained and the reduction in geographical barriers also raise issues.

Social networking

Social networking is one the biggest uses of internet data in the world. There are in excess of 1 billion Facebook users on the planet, and each of them has the opportunity to interact with any of the others. The integration of these networks with gaming platforms has created an arguably artificial society. With the increase in internet relay chats (IRCs) on websites, users are able to talk and interact with several users in separate time zones simultaneously. This can be a positive feature for organisations, as social and technical skills are used in these communications but it can cause an addiction to these types of site.

With the increase in mobile technology, individuals are able to access social networking sites via mobile applications. This means that you do not need to log on to a computer to access them. It also makes them more accessible from the workplace, which is detrimental to productivity and can lead to consequences for the employee if they are caught.

Research

Social networking sites like Facebook have had a big impact on both individuals and organisations. Research both the positive and negative impacts of social networking.

Lack of social skills

Many people blame the increase in social networking and texting for the decline in social skills in younger generations. There are fewer participants in social activities

among the 14 to 25 age group than there used to be. This is, in part, a result of the increase in gaming technology and the access to it from anywhere (mobiles will link with games consoles).

The accessibility of the internet and computers from anywhere has enabled young adults to stay connected 24 hours a day, which can have an effect on sleep patterns and create a high dependency on caffeine-infused drinks. In the long term, there is potentially a higher risk of heart issues and diabetes.

Deficiency in real-world social skills will have an effect when attending interviews and professional meetings. Individuals may not have the necessary skills to be seen as employable to an organisation, which will lead to greater unemployment overall.

Health and age-related issues

Developments in technology are having an effect on the older generation. With technology becoming more and more accessible to everyone, people are finding it easier to use mobile technology. Tablets are a popular choice with all generations. However, although many older people are fairly comfortable with technology devices such as tablets, others struggle with technology and using features, such as small screens and touch keyboards can prove difficult. Older people can also find the rate of change in technology difficult to cope with. This can lead to feelings of frustration and isolation, as older people may find it difficult to communicate with children and grandchildren who are comfortable with new technologies.

People with health-related issues can use the positive effects of technology to keep in contact with medical professionals online. It is possible to use mobile computing to ask for advice and guidance at any time of the day or night, and most doctors' surgeries use an online booking system to make appointments and request repeat prescriptions. Technology is already extensively used in health care and this has improved monitoring and diagnosis in many areas. For example MRI (Magnetic Resonance Imaging) and CT (Computerised Tomography) scanners are invaluable tools for diagnosis. Many advanced countries will have an increasingly aging population in the coming years, which raises issues about how to care for the elderly. Japan for example currently has 25% of its population over 65. Some have suggested that robots may be able help in the care of the elderly, with social companion robots in development.

Maintaining contacts

One positive use of social media today is that people are able to keep up connections with distant family and friends. This removes geographical barriers, as people are able to communicate with the other side of the world at the press of a button. VoIP applications such as Skype® have enabled people to see and talk to someone on screen who may be many miles away. Businesses are able to use Skype® to hold meetings with people in many different locations.

Research

Speak to your elderly friends, neighbours or relatives. How much use do they make of technology (smart phones, tablets, the Internet, Facebook, Skype etc.). What do they like about it and what do they find difficult about it? Do they feel 'left out' by technology?

Employment and business impact

Developments in computing technology have increased home working and have changed the skills needed in a workforce and created new markets and opportunities.

Home working

Home working is a very practical option for the many workers with a good internet connection and a home computer.

Benefits

Employees are now able to access their work from home, which enables them to provide the same services that they would normally perform in an office

This can have advantages and disadvantages. There may be an increase in productivity, as the employee is in a comfortable environment and is able to work at their own pace to get tasks completed. Their work may be more accurate than it would be in a rushed office environment. Further benefits include no time being wasted during the commute to and from the workplace, along with savings in the cost of transport and a reduced impact on the environment.

Disadvantages

A negative impact is a lack of social interaction in the work place, as working from home reduces the need to talk to co-workers and management. It can therefore create an isolated environment. Not being able to bounce ideas off

colleagues so easily may harm motivation and morale may be reduced if the team is spread out in different countries and never gets to meet.

Research

What kind of jobs can be done from home? What kind of IT equipment is needed? Would you like to work from home as opposed to working in a office? What do you think the advantages and disadvantages would be (for both you and the organisation you might work for)?

Low-skilled jobs

With automation and robotics increasingly taking over low-skilled jobs, there is now less of a need for these positions. Robotics and automation will not be able to replace all low-skilled jobs, but the vast majority can be undertaken by these machines. This increases productivity for the organisation that uses these technologies.

However, the organisation needs to employ people with the skills required to control these machines. These workers are more expensive to employ than the previous low-skilled workers, but fewer of them are required so the overall cost of wages is likely to be reduced.

New opportunities

As technology develops, so do the ways in which organisations can advertise their products and services. They are able to grow their customer base. Organisations are able to use the development of web technologies to target certain products at certain individuals, although some see this as spying and an invasion of privacy.

IT and the internet, in particular, have had a huge impact on business and commerce. The internet has allowed the creation of a number of large global companies that did not exist 20 years ago, such as Google™ and Amazon. There have also been casualties – several large multinationals that were household names 20 years ago, such as Kodak, have failed to adapt to changes in technology and are no longer what they once were. E-commerce has grown to the point where traditional high street shops are under pressure, but it has also led to growth in the parcel delivery business, since goods bought online need to be delivered to the purchaser's home address. Services such as Amazon Market Place and eBay have allowed many people to run a business who might never have been able to afford to open a traditional shop. They are also able to sell niche products nationwide (and even worldwide) that could never justify a geographically limited shop. Social media sites such as Facebook and Instagram allow them to promote their services to a much wider audience and at much lower cost than would have been possible with traditional advertisements in newspapers, on the radio or on TV.

Research

Kodak were a technology company who developed the first digital cameras. Find out what happened to them – why did they go bankrupt?

The internet and high-speed broadband connections have also created new opportunities in the entertainment industry. For example, in the past, a new band would have found it difficult to get exposure for their music without being able to sign a contract with a music label. However, today, using sites such as YouTube, bands can upload tracks and videos and more easily develop a following than was previously possible.

Blogging is another example of an opportunity that has been created by the internet. Many people create text or video blogs on a wide variety of topics such as cooking, gaming and health and beauty. Those blogs that attract a large number of followers make money for their creators through advertising revenue.

Environmental impacts

Computer systems have an impact on the environment both from their electricity consumption and from disposal of old equipment.

Electricity consumption

Reliance on electricity is starting to have an ecological impact. Power stations are using more fossil fuels to create electricity, creating pollution that billows into our atmosphere. This is causing problems for the ozone layer and affecting the North andSouthPoles of our planet. This dependence on electricity is creating a need for green and sustainable energy sources. Wind and solar power are becoming increasingly popular with individual households who want to produce their own electricity and save the environment.

Organisations are now starting to build offices that have a green carbon footprint. One Angel Square in Manchester, which is the head office of The Co-operative Group, collects recycled water on-site, to be used in the bathrooms and kitchens. LED (light emitting diode) lighting is used to reduce the amount of electricity needed to light the offices. The electricity used to power the whole

building is created using rapeseed oil, grown on the Co-operative Group's own land and processed in its own plants. Any electricity that is surplus to requirements is sold back into the National Grid.

These buildings show that you can be ecologically friendly and use the Earth's natural resources without polluting the environment.

Improvements in computing power

Traditionally, as computer components become more powerful, they need to use more kilowatts of power per hour in order to be able to run at full capacity. Power supply units (PSUs) are still being developed to take into account the increased electrical power needed by top-end hardware such as graphics cards and processors.

Recent developments in mobile computing processors have placed more emphasis on improving computing power while reducing electrical consumption. The primary goals for this are improved battery life and less heat generation. Many modern laptops can run for more than 10 hours on a single charge. The lower temperature also makes them easier to use on the lap.

Efficiency in energy usage is important in organisations that have a large number of computers, as electricity costs money. This should eventually balance out increased use against the saving in energy.

Waste materials

For each new component that is installed, there will be an old one that needs to be disposed of. Organisations should use safe and environmental procedures, such as recycling, for this, but, in reality, many components will end up in landfill sites. Home users are throwing away more and more electrical goods, which creates more pressure on landfill sites. These will need to cope with the fact that this waste will increase over the coming years as more and more developments are made and more people want to have the newest gadgets.

Recycling

Recycling is becoming a bigger and better way of removing objects from a workplace. Computer components should be recycled by the organisations that discard them. These components can then be stripped down and remanufactured into new components.

Motherboards are melted down at different temperatures to remove the precious metals and the plastics, which can then be sold back into the manufacturing industry.

It is becoming a necessity to recycle dangerous chemicals and battery components as they can cause environmental harm if put into landfill sites. Via recycling, these chemicals can be safely removed from their surroundings and disposed of in a safe and environmentally friendly manner.

Recycling to the third world

With the growth in computing in the western world, organisations want the most up-to-date technology to keep up with developments. Thus more and more components and systems are being thrown away. This unwanted technology can be exported to countries in the developing world.

These countries are able to use this technology to help to improve their skills, even if not the latest technology, so ensuring that they can connect to the modern world.

However, an issue that comes with this amount of technology moving into these countries is the power that it will use. With more and more technology coming into these areas, more and more power is needed to power the devices. This puts a strain on the already thinly spread energy companies. This could drive up electricity prices and make the technology more inaccessible.

There is also a transport issue. This costs money and there is a question of who pays for it.

Ethical issues

Using technology will always raise ethical issues such as lack of equal access, privacy, the potential of artificial intelligence, copyright and terrorism.

Unequal access

The technological age is beginning to create a 'digital divide' between areas that do and those that do not have high speed Internet connections. The Internet brings so much to us in terms of information, communication, education, commerce, entertainment etc. Restricting some people's access to the Internet puts them at a serious disadvantage compared to those who do have access. Although most town and cities in the UK have high speed Internet access, some rural areas do not. The situation is even worse if you consider people in the developing world, where many already face difficulties gaining access to facilities such as health care and education which we take for granted in Europe. Very little of the developing world has access to high speed Internet and so the inhabitants of these areas face the prospect of falling ever further behind those in the developed world. Another consideration is cost. High speed Internet in the UK currently costs around £20 a month, which can be difficult for low income families to afford, placing them at a disadvantage compared to better off families.

Older people and disabilities

Technology is changing at a rapid rate. It is possible that a new computer system will be out of date within months of being purchased. This also means that new skills will need to be learned to fully access the capabilities of the technology. People who have grown up in the 'digital age' are likely to adapt more quickly to the developments, but older generations may not cope so well. New developments usually result in more expensive products, which can be an issue for a pensioner who does not have the money to buy them.

Research

Look into the ways in which technology can be used to assist people with a range of different disabilities.

New technology can create both difficulties and opportunities for people with disabilities. New interfaces such as touchscreens can be difficult for some users, but, on the positive side, stunning developments in cybernetics can transform the life of an amputee. Speech recognition software, sip and puff systems, Braille input devices and adaptations to keyboards are just a few of the developments that enable many people with impaired movement or sight to use computers.

PAUSE POINT Investigate the ways in which technology can help people with disabilities make full use of computers and assist them in their lives.

Hint Visit http://www.washington.edu/doit/videos/index.php?vid=33 to find out about the ways in which computers can be adapted to allow people with all kinds of disabilities to make use of all their functions.

Extend Think of other ways in which you think that technology may be able to help people with disabilities in the future.

Privacy issues

More and more data is being collected about every user accessing an application or website. This creates an issue in relation to privacy and how this data is being used. Many websites store cookies on the visitor's computer and this information can be used to target advertising at the user. A user's internet search history can also be used to target advertising.

All of this data has to be stored somewhere, and this amount of big data can be used by many organisations. This data will be stored on a server that could possibly be accessed by someone with malicious or illegal intent and personal information could be stolen.

Personal data

Personal data is often collected by web sites you buy things from or services you subscribe to. Organisations need to consider how they will collect this data, what data they will use and how it is stored. All this information is personal and needs to be kept securely to prevent any thefts or stolen IDs.

Organisations are required to say clearly how they will be storing the data they collect. If they use cookies that collect information, this needs to be clearly visible to the user, who must have the option to disallow them.

Legislations

The UK has laws designed to protect intellectual copyright, misuse of computers and personal privacy.

Copyright

The Copyright Designs and Patents Act 1988 is there to protect against the misuse of information (audio, video, imagery, literary text and more). It is there so that people are not able to use the information without the creators' permission or

proper referencing and payment, if necessary. Any kind of information that you gain from another source needs to be referenced and you must seek permission from the creator if you want to use their material. Some software is available under licenses such as GNU Genernal Public Licence (GPL), which is essentially copyright free, and users are free to copy and modify the software.

Computer misuse

The Computer Misuse Act 1990 was brought in to protect against the rise of hacking in the late 1980s and early 1990s. It recognises the following four offences

- unauthorised access to computer material
- unauthorised access with intent to commit or facilitate a crime
- unauthorised modification of computer material
- making, supplying or obtaining anything which can be used in computer offences.

This piece of legislation is in place to protect organisational systems against hackers and the threats that they could pose to personal data stored on the servers.

Data protection

The Data Protection Act 1998 protects the personal information of all individuals. This law was put in place to control the way that this information is kept and handled. Data cannot be passed between organisations without the permission of the individual involved. Individuals have the right to say that organisations cannot use their information and organisations must respect and honour that wish.

Any information about an individual can be covered by the Data Protection Act. The types of information that organisations are most likely to want to capture and store include:

- names
- addresses
- contact information
- convictions
- medical conditions
- credit history
- employment history.

Other European nations have similar laws in effect, as personal information can be stored in multiple nations.

Risks from data stored online

There is a risk in today's society that anything that is stored on the internet stays on the internet. It only takes a second for an image to be posted and then saved and shared. Attacks on major companies such as SnapChat and iCloud have prompted organisations to make sure that all personal information that is posted online is kept securely.

There is always going to be a risk to personal information posted online, and this is widely understood and accepted. Organisations have a responsibility to ensure that data is kept safe, and therefore they need to spend money on and constantly update the security measures that they put in place to protect this information. Organisations that use cloud storage need to be aware of a number of risks associated with using the cloud. Often the cloud storage is provided by a third party cloud storage provider which is not under the control of the organisation that owns the data. There need therefore to be careful checks that the cloud storage data provider is trustworthy and applies security measures rigorously. With cloud storage the organisation that owns the data may not know exactly where the data is stored; it could for example be stored on servers in a different country, where different laws apply.

Artificial intelligence

Artificial intelligence (AI) is one of the huge growth areas in computing and the technology behind it is becoming more commercial. There are positives and negatives behind all technology advancements, but there are growing concerns about AI. There are potential dangers involved with AI technology in certain industries. Handing over complete control of a manufacturing floor to AI devices could cause issues when humans want to update or check on progress.

However, the use of AI technology is becoming popular in many areas, from manufacturing to business. Some typical applications of AI include:

- financial predictions such as attempting to predict stock market price increases or decreases or assessing the ability of a bank customer to be able to pay back a loan.
- Computer vision, used to make sense of photos or video data, which in turn can be used for robotic systems (such as a self driving car), satellite photo interpretation (in military systems looking for enemy movements).
- Speech recognition, to allow computer systems to accept verbal commands.

Without strict protocols there is the potential for the AI devices to develop further after creation, so that these devices could develop towards being able to outwit human intelligence. This could, in extreme cases, pose a risk to human life.

Robot weapon systems

The military have been developing AI systems for a number of years, but this comes with dangers. Having

fully AI machine-controlled weaponry is dangerous, and there needs to be a manual override on the machine. These weaponry systems have control over targets and firing power. Consider what would happen if a malfunction were to happen and the machine aimed at a 'safe' area where civilians were living. What would be the repercussions for the nation in charge?

Negative aspects of internet use

The internet is a powerful tool. You are able to find anything on the internet if you use the right search words, but there are also many negative aspects associated with the internet. The following are some of the recent developments associated with this.

- Cyber-bullying – This is bullying performed online. There have been many cases of individuals bullying others on the internet for no apparent reason, except for the fun of it. Ask.fm has been in the news over 'trolls' and bullies abusing random members of the website. The effects of this can lead to the person being bullied committing suicide and there have been several cases of this reported recently.
- The 'Darknet' – This is a version of the web that has been developed and used by criminals and terrorists. It is a network where criminals can communicate privately. These areas can be used for a recruitment drive, for posting messages and for providing chatrooms on the internet with the purpose of recruiting people to a terrorist organisation and communicating their propaganda.
- Pornography and revenge porn – Pornography is a huge internet business, and there are millions of websites dedicated to this industry. Each one has the potential to make money. Revenge porn is illegal in most countries, and there have been many reported cases of convictions happening because of this.

Research

Bullying has always taken place, but how has the internet facilitated it? Find some recent example of how cyber-bullying has taken place and what the impact was. What is being done to try and prevent it?

Illegal material on the internet

Material that can be found on the internet is not always free to use or free to view. YouTube is a website for uploading and viewing videos, and much of the streaming is illegal because videos can be uploaded that are in breach of copyright law. Despite warnings, many people continue to upload this material even though their YouTube accounts may get blocked as a result.

Peer-to-peer (P2P) file sharing became popular in 1999 when Napster came online. This system kept lists of music held on the computers used by the members so that copies of music could be transferred between these personal computers. Napster stopped the music sharing service in 2001 as this activity breached copyright law.

Video streaming web sites, while legal in some cases, can provide illegal access to television stations that are normally premium channels on satellite television. These websites are banned under EU law, but are mainly hosted in other regions where laws can be more lenient.

Use of the internet by terrorists

Personal freedom always has the potential for abuse. We expect to be able to make personal and private phone calls, texts and emails, but this freedom can also be used by people conspiring to commit acts of terrorism.

PAUSE POINT There is a growing need to screen the big data obtained from email and mobile communications to identify possible terrorist threats. What are the implications for our personal privacy and freedom?

Hint Can you find a news story about security services monitoring terrorist communications?

Extend Do you think there are any real concerns over the security services monitoring the communications of law-abiding citizens?

Assessment practice 9.1

A.P1 A.P2 A.M1 A.D1 B.P3 B.P4 B.M2 B.D2

You work as a junior member of the programming team for RR Holdings, a large national company involved in property lettings and management. Your team leader, Tamas, has been asked to produce a presentation and supporting documentation to explain the impacts of computing systems.

The audience for the presentation will be the senior management of RR Holdings, who wish to make the company more environmentally friendly.

The presentation needs to include:

- impacts that developments in computing have had on an organisation
- likely impacts of an emerging technology on organisations
- potential ethical and environmental impacts of developments in technology
- how a lack of understanding or access to IT can disadvantage certain groups of people.

The supporting documentation needs to include:

- risks related to implementing a new computer system in an organisation
- evaluation of the impact that implementing a new computer system can have on an organisation
- benefits and disadvantages of the social impact of computing technology developments
- impacts that the implementation of a specific development in computing technology has had on wider society.

Plan

- What am I being asked to do?
- What are the success criteria for this task?
- Are there any areas I think I may struggle with?
- How confident do I feel in my own abilities to complete this task?
- Do I have any existing knowledge about the task at hand?
- How much time do I have to complete the task? How am I going to successfully plan my time and keep track of my progress?

Do

- Have I spent some time planning out my approach to the task at hand?
- What strategies am I employing? Are these right for the task? Are they working? If not, what do I need to do to change this?
- Can I identify when I've gone wrong and adjust my thinking/approach to get myself back on course?
- Do I understand when to consider and when to be decisive (reflection vs action)?
- Can I improve my own learning approach?
- Can I improve my own learning environment?
- Am I utilising all of the support available to me?
- What am I struggling with? Do I know how to overcome this?

Review

- I can explain which elements I found easiest.
- I can explain which elements I found hardest.
- I can describe my thought processes.
- I can draw links between this learning and prior learning.
- I can explain what skills I employed and which new ones I've developed.
- I can explain what success looks like.
- I can use this experience in future tasks/learning experiences to improve my planning/approach and to monitor my own progress.
- I can identify how this learning experience relates to future experiences (i.e. in the workplace).

C Develop a plan to implement a computing technology development in an organisation

As part of your assessment for this unit, you will need to develop a plan to implement a technology development within an identified organisation and to manage the associated risks.

For example, you could consider the rollout of new accounting software in a pharmaceutical company. This example offers a lot of scope for your plan including methods used to input data, how to safely move the accounts from the old system to the new, as well as the risks and consequences of a hacker accessing the data.

Information gathering

The first action to take before implementing a new system is to research whether the system is viable.

Investigation of the planned implementation

When starting to plan the implementation of a new development, you need to investigate how this implementation will be conducted, who will need to use it and how will it be used. Investigation into these plans is a key part of deciding the way in which the development will happen in order for it to be successful. This planning should also consider timescales and costs.

Sources of information

Being able to identify which forms of information are essential for the development is important for ensuring a successful implementation. Some of the possible forms of information and their sources are given below.

- Images – Drawn or captured imagery can help to design the development, which could be either the user interface (UI) or the context or location of the development.
- Journals – Academic documents can give information about past developments that have been similar to your planned development.
- Websites – These can provide data or information that can provide guidance or give answers to questions that may arise during the planning of the development.
- People – these could include the users of the existing system that the new development will replace and other people with an interest in the development such as the managers or owners of the business who have requested and are paying for the development.

Questions, queries and searches will determine which sources are the most beneficial for a particular context.

Information gathering techniques

There are several ways in which information can be captured and used inside an organisation.

- Interviews – small formal conversations, with targeted questions aimed at getting certain guided answers.
- Meetings – which can be formal or informal conversations around certain given topics aimed at gathering information from all members and coming to a possible consensus.
- Observation – a single person or peer watching an activity to gather information about how something is done or performed.
- Data analysis – when big data is captured, it will need to be analysed and interpreted into information that can be understood by the organisation requesting it.

Each of these types of information can be analysed and used by anyone associated with the organisation, which may include employees, management, suppliers, or the general public and customers.

Using these methods of information gathering can duplicate data and cause issues for the analysis of the information. Organisations and those conducting the data gathering will need to make sure that they use the appropriate methods for gathering this information. Using the wrong method can give negative results and render the analysis useless.

There can be positive benefits gained from asking the public for opinions as they are likely to represent people who are involved with the new system. For example, if a bank were planning to close all their high street branches and just use mobile and internet banking, this could have a big impact on the members of the public who are their customers. You might therefore want to carry out a survey among members of the public as you want their views on the subject. Members of the public may have ideas and views which were not anticipated but which need to be considered to help make the new system successful. However, surveying the general public can create problems. You need to design a questionnaire to collect the information you require, and you will need to survey a reasonably large number of people, otherwise your results may be biased. You also need to be careful about the people you choose for your survey. For example, if you asked younger people about high street bank branch closures, they may be quite happy just to use mobile and internet banking, but if you asked older people the results might be rather different. Therefore carrying out a proper survey of public opinion is time consuming, because it involves designing the questionnaire, collecting sufficient responses and analysing the results.

By gathering data correctly and using proper forms of analysis, you will be able to use the information to make a positive impact on the organisation and its profit figures.

Each method of gathering information has its own benefits and drawbacks. Interviewing people inside the company can be a good way of finding out how they currently work, but you need to ensure that you speak to the right people to get accurate information. It can also be a time consuming process and some people within the company may not have time to be interviewed. You should also bear in mind that you may get different answers depending on who you interview. For example, a manager within a company might tell you how a process is supposed to be completed but the person who actually carries out the process might tell you a different story. This is where observation can be useful. Since you are just watching the activity, you should not need to take up so much of people's time and you can see how people actually carry out the task.

Implementation planning

You need to be able to plan how your computing technology development will be implemented, including the scope of the project, constraints, timescales, milestones and how it will be monitored.

Requirements analysis

A very important aspect when planning a project is the process of determining the expectations from both the end users and management (who may not be users but have an interest in the development). This is known as requirements analysis. This process involves talking to the existing users and determining what they want from future developments, and then using this information to form the basis of the new development. Clearly, it is important that the end user is satisfied with the final product that is developed.

When gathering the requirements, they must be:

- quantifiable
- relevant
- detailed.

Each of these is important when planning the development as the more detail that the end user provides, the better the planning can be to meet the user's needs. This may involve assessing the needs of several users as it may not be just one organisation that uses the previous technology.

Intended achievements

You must ensure that whatever you have designed or updated is done to achieve a purpose. Does the new product do what is expected of it? There should be a clear plan of what the product should and should not do and throughout the design and development of the product you will need to refer back to this plan to ensure that the original intention and requirements are going to be achieved.

Scope of the implementation

Identifying the ways in which the project will be successful is a key part of planning the development and setting the targets that need to be met in order to achieve this. Scoping the project means formalising the goals that need to be achieved.

This starts with the overall objective of the implementation: that is, what is the end product supposed to do? Breaking the objectives down into smaller and achievable goals will make the development seem more achievable to the organisation.

The scope must be agreed by the whole team working on the development, so that everyone understands how to make this implementation work. Although it may be tempting, you need to make sure that you do not add anything extra that has not been requested by the requirements analysis.

Boundaries and constraints

With all project developments, you will have to work within certain boundaries and constraints, which can range from costs to compatibility with existing systems.

- Cost – Your project will be given a budget that you must stick to. You will be asked to make sure that everything that is associated with your project is documented and costed so that the whole development is cost effective and will not go over budget.
- Timescales – Organisations will need to make sure that all developments are completed within a certain time frame. You will need to work out how long everything will take, and charts such as Gantt charts (see Figure 9. 4) are used to determine the longevity of a project and the predicted duration of each project task.

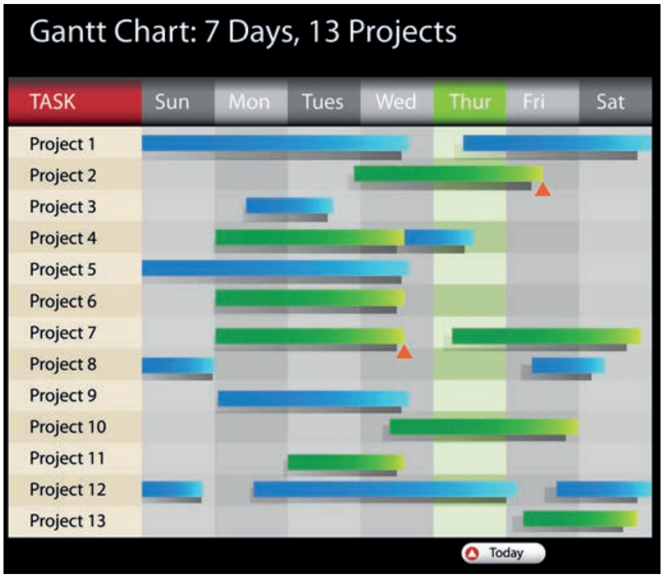


Figure 9.3: Gantt chart

- Hardware platform – When planning a new development either in hardware or software, the developer will need to make sure that it is compatible with existing systems already on the market and inside the organisation. However, ensuring that the new development will work with existing and newer versions can be a challenge. Consideration also needs to be given to the minimum hardware required to run the product, including specifications for processor, RAM and disc space.

Each of these can cause problems when planning and developing a new technology or component. Being able to see the constraints before they happen and having a plan in place to minimise the impact will help to make the development run more smoothly than if no plan was made.

Inputs, outputs and processes

You need to identify the system's basic inputs, outputs and processes and to consider alternative solutions.

System requirements

When developing a new software application consideration must be given to the hardware and software resources required to run it. With a mobile app you will need to decide which mobile platform(s) it will run on, for example iOS, Android, Windows® mobile etc. and which versions of the operating systems. For applications designed for desktop computers you will also need to decide on the operating system(s) required (e.g. Mac OS, Windows®, Linux) and versions supported. You will also have to decide hardware requirements such as disc space and RAM required. With web based applications you will need to decide which browsers and which versions of those browser will be supported. While supporting a wider range of hardware and software might seem like a good idea because it will widen the appeal of a commercial product, it will mean the development and testing of the software will take longer.

How the development will work

Flow charts are an easy-to-follow graphical way of showing what needs to be input and the processes that need to be undertaken to achieve the intended output of the new system. The development team will need to identify what information will need to be put into the system, and the way in which this will be achieved.

The way in which the information will be processed, and how the output will be presented needs to be agreed upon at the start of the development process. This may involve a simple process, such as addition of the input data, or be more complex, such as taking the input, analysing it and producing a graph or table to show how the analysed data depends on the queries added.

Alternative solutions

The development team will need to make sure that the process they agree on is the best for the development and, to achieve this, should consider all alternative ways for it to be carried ou,t along with their merits. For example, is there a way of reducing the processing power that is needed, to ensure that the solution is more economical for the system, the power consumption and the environment?

Implementation plan

As you start to create the implementation plan and you have your objectives set by the organisation, you will need to create a plan with **SMART** targets and milestones. These should be small enough to be achievable and timely so that they are achievable within a set amount of time. These milestones or tasks should be detailed enough to show exactly what must be achieved by each of the deadlines.

These outline plans should state all targets that need to be achieved to complete the final objective. Each of these tasks can be small and last for just an hour, or could be larger elements that take longer. This is why setting SMART targets and monitoring your progress as you move through a project is important for achieving the end user's set goal.

Key terms

SMART – Short-term Measurable Agreed Realistic Time bound targets (SMART) are widely used for agreeing targets. An important aspect of this is 'measurable': that is, to be able to clearly identify whether a target has been met or missed.

Monitoring progress

As shown in Figure 9.3, the Gantt chart can be used to monitor progress. As each task is completed, it can be coloured in and part tasks can be shown by using enough colour to indicate how much has been completed.

Milestones can be entered into a diary, checked and documented as the project continues.

Gantt charts

Gantt charts are useful here for showing how long each task will take and when it will need to be started. Some tasks cannot be started until a previous task has been completed. Having a visual form of this can be helpful to show these dependencies and get an overview of the timescale of the whole project.

Target sheets

These can be in any form, from a simple checklist to a detailed table which enables you to give basic information about the task set and the date by which it should be achieved. It would be helpful to have a central target sheet, held by the development leader or the stakeholders, and then smaller individual sheets for each member of the team detailing their specific tasks.

It is essential that careful monitoring of the progress of these tasks is carried out to make sure that the tasks and milestones set by the whole team and those given to individuals are being met. Making sure that these targets are realistic will be the responsibility of the team leader or the stakeholders of the organisation, and they may need to adapt the tasks or the deadlines to make them more achievable if it seems that they are going to overrun.

Case study

Your friend volunteers at a small local zoo and has mentioned to you that they currently have a paper-based scheme where members of the public can sponsor a particular animal at the zoo. They would like to move the sponsoring process onto their website and link it with their Facebook page. You arrange a meeting with the fund raising manager at the zoo and he explains how the current process works. People who are on the zoo's mailing list get occasional mailings sent to them with details of some of the animals available for sponsorship and how to set up the process (select the animal and complete a monthly direct debit from their bank). Visitors to the zoo are also given a flyer promoting the sponsorship scheme. Once people have joined the scheme, they receive quarterly mailings telling them about the animal they have sponsored.

The fund-raising manager tells you that she hopes an online system will reduce the cost of mailing and also make the updates about the animals more interesting, for example by including videos. She also hopes that using an online system will help to promote the scheme, via the zoo's Facebook page, and attract more people to take part. You also contact a number of the existing sponsors to see how they would feel about an online rather than the existing paper-based system. The new sponsorship web pages will need to be integrated with the existing website. This is hosted by a web hosting company (who do support web programming standards such as PHP and MySQL) and developed by a volunteer who is a professional web developer and does updates in his spare time.

The basic process for the sponsorship scheme will be as follows.

- On the existing 'Photos of our animals' page on the website, a 'Sponsor Me' button will be added against each animal that is available.
- Clicking the button will take the user to the 'Register' or 'Log-in' (for existing users) page where users will enter their registration details.
- From the registration page, they will be taken to a page where they need to fill in their bank details on a form to create a direct debit to sponsor the animal they have chosen.
- Once the user is registered, they will receive regular email updates including photos of the animal they sponsor.
- Sponsors will also have access to a special area on the website where videos of the various animals will be uploaded.

The outline implementation plan you agree with the web developer is as follows.

- Develop designs for programming, the database and the user interface (web pages) – 4 weeks.
- Agree design with zoo management – milestone.
- Develop database, programmes and web pages – 6 weeks.
- Test system – 5 weeks.
- Demonstrate final system and obtain agreement to release the system with zoo management – milestone.
- System goes live – milestone.

Consider the following:

- What additional information do you think you would need to gather about this project to complete the planning?
- What do you need to add to the outline of the implementation plan to make it a fully completed plan?
- What sort of risks are likely to be associated with this project? How would you manage them?

Create a questionnaire you would send to exiting animal sponsors to find out how they feel about an on-line version and what they would want to see included.

Managing risk

There are always risks involved in a project. The best projects identify these and take measures to manage and reduce the impact of them.

Planning

With all new technology that is being developed, there is an element of risk. Planning to reduce these risks can be a challenge, but knowing how the development will work can help you to identify any risks that could be associated with it.

Having a plan in place before you start, with potential risks already identified, will help during the development, as the developers will know how to combat any risks that do happen. This will not be a definitive list of risks as more can develop as the development process is implemented, so these plans will need to be updated and changed as the development progresses.

Some risks cannot be controlled by the development team, for example competitors can start to develop a system that is a rival to yours. Being able to adapt as times change and systems improve will need to be part of the risk management process. Do you add more functionality, or make the system more efficient?

Training

Development of the staff on the newest technologies is essential for making sure that the organisation can run with high efficiency and productivity, but this can come at a cost. The higher cost would arise from the need to provide training for the workforce by the company that developed the software or system. These costs could be lowered by using a member of the in-house staff team to train their peers in the new software or system.

Users

User training is important to ensure that all users know what the system has been set up to achieve and how to use it. Sadly, a lot of organisations expect users to find their own way around a new system. A much better practice is to educate staff so that they can make full use of the new technology.

Support staff

Support staff should receive training so that they understand fully how the new computing technology development works and, in turn, are able to help their users with problems they encounter.

Support mechanisms

Support keeps the users productive and reduces frustration due to problems in making parts of the system work.

Online support

Support for all aspects of IT can be found online, and many companies have set up support online for their customers or clients. Online chat enables users to interact with the company without picking up the phone or walking into a store. This form of messaging allows not just text communication but also other media such as files, screen shots etc. of the problem can be sent. Customer service/support personnel can also run several chat conversations at the same time. This helps improve the productivity of the customer service department.

Helplines

Support can also be provided using a telephone helpline. This has the benefit of being available even if online help is not (perhaps due to the type of problem for which help is being sought). However it does require a dedicated support person for each call, which can mean long waits for queries to be answered at busy times.

Outsourcing support

Outsourcing support is a good option for organisations who prefer to specialise in their core business and to pay another company to provide the support for the system.

There are many types of outsourcing contracts, ranging from remote telephone support to staff from the outsourcer having a permanent on-site presence.

Quality assurance procedures

As the development process progresses, the development team need to make sure that the product they are developing is of the highest quality and to the standard that the end user requires. Quality assurance procedures throughout the whole process can help to make sure that everything developed so far is doing what it should and that it is compatible with the existing systems and can function without errors.

Quality assurance could start with a technical review of the systems that are already in place to see whether they will need to be upgraded, if the organisation can afford this, and whether the development will need to have a lower minimum expectation.

It is important to review the project at all stages to make sure that it is meeting the targets set and that development is progressing according to plan. These reviews can identify ways in which to facilitate better and faster progress and possibly enable the development to get ahead of deadlines.

Major developments will have been made during the course of the project, and each individual element should be tested thoroughly. Then these modules should be combined and tested as a group to make sure that they all work together correctly. This is known as integration testing or I&T. Once this has been carried out, a validation test should occur to make sure that everything is acceptable for the client and end users and fully meets their requirements.

Disaster recovery

Disaster recovery plans should be in place to protect against a major breakdown in the system. This could be a simple system error or a whole network problem. The need for this kind of plan was recognised in the 1970s as

organisations started to see their dependency on computer systems increase and wanted to be able to continue working as quickly as possible after a disaster, with as little impact as possible on the organisation.

Disasters can take many forms, for example a flooded server room, fire, theft, building collapse or hacking. The best disaster recovery plans involve practising restoring the systems onto new computers to confirm that the recovery plan can be relied upon.

These plans will involve policies and procedures that must be carried out to make sure that the data stored on the servers is kept intact as much as possible. This will enable the development team to continue working without losing too much data.

PAUSE POINT Why would an organisation want to outsource IT support?

Hint Research an outsourcing company to find out the services they offer.

Extend Identify a scenario where outsourcing would clearly be a good option and a scenario where this would be a poor one.

D Review a plan to implement a computing technology development in an organisation

The most effective plans to implement change have a review phase to ensure that all went well and that any opportunities to improve the end result are identified.

Obtaining feedback

Feedback is a powerful technique as other people often have a more realistic view of a system than those who were involved in setting up the development.

Reviewing plans with others

Once a plan has been created and written, it is a good idea to get this plan reviewed by the members of the organisation who will be involved in making the development possible. Stakeholders will discuss the plan with them, and possibly suggest any improvements that need to be made in order to make it viable for the organisation.

It is also good to get these plans reviewed by someone who is an expert in the subject area of the specific development. This expert will know how the development could pan out and may be able to suggest alternative ways for designing or implementing it.

This expert would help to ensure that the development that you want to implement is both possible and achievable in your given timeframe.

Feedback methods

Gaining feedback for an organisation is a valuable exercise that can be used to further develop a product or idea. Organisations will gather the feedback of their users through multiple platforms, enabling them to gain a wide mix of comments and opinions.

Interviews

Interviews are the most time consuming option as they involve a member of the team being directly involved in gathering and recording the feedback. This technique is very powerful and can return good value for the time involved, as a skilled interviewer is able to ask supplementary questions to probe and understand the feedback received.

Questionnaires

Questionnaires are a quick and cost-effective method of gathering feedback from a lot of people. However questions need to be carefully phrased in order to gather accurate information. Also people may give what they consider to be the 'correct' answer rather than a truthful answer.

The questions can be quantitative where a yes/no or number response is entered. These questions are a good source of data for producing graphs, charts and percentages.

Open questions can be useful to find out new ideas and suggestions from the sample.

Meetings

It can also be useful to gather a demographic group of potential users to gain valuable feedback in the style of a forum. Members are able to freely discuss their thoughts and critique the development. This will give the organisation feedback that they can use to develop their product for that given demographic. However meetings take time, to organise the meeting, to take part in it and analyze the feedback gathered.

Online surveys

Online surveys, normally used as 'pop-ups' when a user lands on the home page of a website, can be a valuable way of gaining feedback. Users will be able to give constructive feedback either anonymously or named, so giving the organisation a wealth of data that they can use to develop the product further according to their customers' wishes. However the return rate on this kind of survey can be very low as many people do not have the time or motivation to complete them.

Ratings

Mobile platforms use this as the main form of feedback for a game or application. A simple star rating will give the organisation/developers an idea about how the application is valued and whether more detailed feedback is required.

Researching similar projects

With all projects that are undertaken, no matter how big or small, it is a good idea to begin by researching into similar products or projects that are already on the market or have been implemented. Market research is not just about seeing what is out there, but will help you to decide how your product/development can be different from those that already exist. The information gathered here has to be used positively to either reinforce the ideas that you already have or inspire you to find a way to be different.

Thorough analysis of what already exists can help you to identify gaps in the market that your new development can fill. You can also compare a completed product to similar ones in the market place, to identify the good things about the product that has been developed and also how it might be improved in future.

Once this research has been completed, you can start to consider how your development will fit into the market. Will it be similar to others or will there be a feature that is different? If so, how will it be different?

Review and analyse

After feedback has been collected it will need to be reviewed and analysed. This is important for summing up the development and suggesting any improvements that could be made in the future.

Reviewing feedback

Each piece of information gathered can be used to assess how well the development is progressing and how it can be improved as it moves forward. Feedback can be gathered throughout the development – it does not have to be only given at the end. Feedback throughout the project can help to adjust the development as it progresses to keep it on track and responsive to new ideas. It can also help you draw conclusions about how well the development process went and how you might improve it next time by doing some things differently.

Seeking feedback from reviews is beneficial because other people often see flaws and good practices not noticed by those actually involved in the project. Using reviews should make the outcome of the project a better match to the client's needs.

Reviewing impacts

The impact that the development will have on wider society, employment and the environment should also be considered during the review process. In particular, reviewing how the development will affect the environment is vital for helping to develop a more efficient product that uses less power.

Employment issues and the impact a new development will have on current staff can be important considerations. The existing staff may need retraining to use the system, more staff may need to be taken on to cope with anticipated growth, or fewer staff may be needed as the new system will automate some tasks currently done manually.

Some systems which interact with the public may have an impact on wider society. For example banking systems

have changed significantly in recent years with many people now managing their accounts on-line.

Privacy issues

It is very important to make sure that plans are in place to protect all data when any technology is developed, and how this is carried out is up to the development team. In doing this, they must follow strict guidelines to ensure the privacy of the data.

Data Protection Act

The Data Protection Act (1998) ensures that all personal information for an individual is protected. Any development will need to put a protocol in place to make sure that personal data is not used against an individual's wishes. If an organisation wishes to use personal data, they must have the permission of the individual involved first and processes must be put in place to obtain this when the data is collected. This is normally by means of a tick box. The individual must be made aware that by ticking it, or not ticking it, they are allowing the organisation to use their personal data. Any development that stores personal data will also have to ensure it is kept secure and complies with all the Data Protection principles.

Protecting personal data

An organisation must make secure storage of personal data a priority to ensure that it is sufficiently protected from criminals, including hackers.

This is not a one-time process, and security measures need constant updates to ensure the highest levels of security. If this is not done, the organisation is more likely to be attacked and have data stolen. This level of security normally comes at a high cost to the organisation, but this is a small price to pay compared with the repercussions of a huge security breach.

Impact assessment

It is important to assess the impact that any new development may have on the existing security system. This will need to be established so that the organisation can put procedures in place to combat any potential conflicts.

Other factors, such as whether the development will require more processing power and the impact that this extra energy consumption will have on the environment, needs to be assessed. Plans should be put in place to enable the development to use as little power as possible to produce the desired results.

Risk assessment

Risk assessments are a part of every aspect of life, ranging from a risk assessment before a school trip to a risk assessment of Google™'s server room. In the context of a computer system development, a risk assessment is needed to identify any vulnerable areas of the system or potential dangers that could arise outside the system, such as criminal or terrorist attacks, and outline the steps that should be taken to ensure that these do not happen. Risk assessment should also look at more mundane events such as a key person on the development team being off sick for a week or more or essential hardware deliveries being delayed. These kind of events are much more likely but can have a serious impact on a project. Contingency planning for these types of events can include building additional time into the plan to allow for delays and investigating possible alternative sources of manpower, such as using external contractors.

This risk assessment should be combined with a rating scale to determine the likelihood and threat level of each potential risk. These assessments will need to be regularly updated to keep up with new developments and system changes.

Skills, knowledge and behaviours

The last part of the review looks back at your performance to identify the positive contributions you have made as an individual and to suggest any potential areas for improvement.

Planning and recording

Planning is a key aspect in any development. Organisations want to know what you aim to achieve and by when you aim to have it achieved by. Target setting must always be SMART (Specific Measurable Achievable Realistic and Time bound); this enables you to know what you have to do and how you have to do it.

One aspect of the planning process will involve gathering feedback from the organisation to find out whether your targets are acceptable, and from the rest of your team to ensure that they are realistic and you have allowed sufficient time. You could even seek the opinion of a target audience. Feedback will inform the way in which you move forward and how you will confront any potential issues that may arise.

All of these targets and issues will need to be made available to all involved, because all members of the team will need to work together to achieve the end goal.

Reviewing and responding to outcomes

Your initial plan will not be set in stone, but can be used as a working document that changes as the project progresses. It may be that some targets are deemed unrealistic once work is started or during the course of a task. Targets and plans will need to be constantly re-evaluated to make sure that no-one in the team is cutting any corners and that targets are going to be met within a suitable amount of time.

As you move through the planning and development stages, issues may arise that will need to be dealt with. These will need to be responded to by the appropriate team members, and plans may need to be readjusted as a result.

Responding to the feedback gathered from others is a very important part of the planning stage and should continue throughout the development process. How others feel about your plan and development will need to be taken into account and acted on so that the plan is not unrealistic and is achievable by all involved.

Your behaviours and their impact on outcomes

How you act when taking part in the development of a project can have an impact on the final outcome.

Professionalism

Being professional throughout the project will ensure that your aspects of the development are completed on time and to an acceptable standard. Your conduct should always be polite, confident and professional. Did you plan your time and then monitor your progress against your plan?

Etiquette

You should be able to understand and practise the different types of communication styles involved in the project with appropriate etiquette used for each. For example if you needed to interview or e-mail some people as part of your research did you use appropriate communication styles and wording?

Supportive of others

Professionalism is not just about doing a good job. You are expected to support your peers if they are in need of assistance or guidance.

Being polite with superiors and any employees whom you lead will help to ensure that you can achieve the best possible outcome for the development and not have a disruptive atmosphere that could affect the timescales set in the planning stage.

Timely and appropriate leadership

There will be many opportunities to demonstrate leadership during a project. The best project leaders always have an understanding of the timescales involved and how individual tasks have an impact on the delivery.

Timely intervention can be very useful to identify when a task is falling behind and may affect the overall delivery. Allocating more resources to such a task can bring it back on track. For example there may have been times when you fell behind with a plan in which you were involved. How did you manage this, what actions did you take to get yourself back on track?

Accountability

The responsibility for your tasks is solely your own, and you must make yourself accountable for your actions. If you are falling behind and not completing tasks, you must own up to it and accept responsibility and any help that you may need.

Teams are about working together and not on your own. Remember: 'There is no I in team.'

Evaluating outcomes

Evaluating the end result (outcomes) of each stage and the final development can help when justifying your methods of working to the client and the stakeholders. At the end of each task, it is good to evaluate the task and suggest any improvements for future tasks. This may help when making decisions about the development and how it progresses.

Once these outcomes have been evaluated, the team is able to make recommendations to the end user and organisation about how the development should work or the systems it needs to run on. Each of these will have an impact on future development of products.

Communication skills

IT professionals need good communication skills to be able to keep management and users informed of planning and progress.

Conveying intended meaning

When communicating with the clients or the stakeholders, it is important you use a form of communication that is appropriate for the message you intend to deliver. Written communication can be used to express your message in a formal or informal way. A brief written note (a memo, for example) can deliver a short message with little detail. Long pieces of writing, such as reports, have the ability to communicate detail along with a clear message.

Verbal communication can also convey a formal or informal message to clients or organisations in one-to-one or in group situations, depending on what is being conveyed.

Use of tone and language

When delivering a message to an audience, it is important to use a positive and engaging tone of voice and appropriate body language as this will help the audience to focus on what is being said. Using language that is understandable and not too technical will enable the audience to follow what is being said, while any complex jargon and obscure language will switch off the audience as they will not understand what is being delivered.

Responding to the contributions of others

Reacting to the contribution of others is important for the morale of the development team. Being supportive of others when they need assistance, responding to any objections made and resolving conflicts can have a positive effect on the team.

Being an effective leader involves making sure that each member of the team has an equal opportunity to contribute to the overall project, and ensuring that everyone is meeting their given expectations. If someone in the team is not doing this, it is your responsibility to ensure that this individual understands the importance of their role and what they do in the team.

All of this can have a constructive and positive effect on the team, so that the development will move smoothly.

PAUSE POINT What makes an effective leader? What skills and other characteristics are important?

Hint Research a leader whom you admire and identify their main achievements.

Extend How do you think they achieved their goals? What communications skills did they use best? How could their performance be improved?

Assessment practice 9.2

C.P5 C.P6 C.M3 CD.D3 D.P7 D.P8 D.M4 D.D4

You have to complete an aptitude test as preparation for a major job interview for a progression position at your place of work. Your test is to develop a plan to implement a computing technology development in an organisation of your choice.

Your plan needs to incorporate relevant documentation such as annotated diagrams and evidence of feedback from others.

The plan will have these sections:

- Information from a variety of stakeholders to explain the potential impact of your suggested computing technology development implementation on the identified organisation.
- Planning to implement a technology development within the identified organisation and to manage the associated risks.
- Analysis of the scope, boundaries and constraints of your implementation plan.
- Use of feedback to evaluate your plan and the suggested improvements.

You are to review your plan under these section headings.

- How well your plan was implemented, considering feedback from others and identifying possible improvements.
- Potential social impacts of your plan.
- Justification of the choices made to manage the risks associated with the computing technology development implementation.
- Your individual responsibility and effective self-management in the development and review of your plan.

Plan

- What is the task?
- What am I learning? Why is this important?
- How will I approach the task?
- Do I need clarification around anything?
- Do I need to be working with anyone else? If so, what is my role? What am I contributing?
- What resources do I need to complete the task? How can I get access to them?
- What aspects of the task do I think will take the most/least time? How will I balance these?

Do

- Am I confident that I know what I am doing and that I know what it is I should be achieving?
- I understand my thought process and why I have decided to approach the task in a particular way. I can explain this reasoning when asked.
- Can I set milestones and evaluate my progress and success at these intervals?
- Can I seek the opinions of others?
- Am I open to change?
- Can I make connections between what I am reading/researching and the task, and identify the important information?
- Am I recording my own observations and thoughts?
- Am I recording any problems I am experiencing and looking for ways/solutions to clarify queries?

Review

- I can explain what the task was.
- I can explain how I approached the task.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).
- I can describe what strategies I employed to cope with failure and self-esteem.
- I can make informed choices based on reflection.
- I can accept that I am responsible for my actions.
- I can explain where I learn best (environment).
- I can explain what I have learned and why it is important.
- I realise where I still have learning/knowledge gaps and I know how to resolve them.

THINK FUTURE

Kerri Morgan
Project manager

The best part of my job is the variety of tasks my team are set and the tasks involved in bringing new technologies into the workplace. A large part of this is the meetings I need to attend with other managers, who are involved with user departments at different levels, to explain our plans and to listen to their feedback. Hearing what other people contribute to our meetings is a lot more difficult than it sounds as we tend to get caught up in what we each thinks is possible and the best way forward.

My team consists of four programmers who are each very impressive in their own ways and respond best to individualised communication methods. Jondel likes a written plan with good detail which he can follow and tick off. Tangel prefers ad hoc conversations to remind her of the current short-term goals and to report progress. Ahmed is very visual and likes diagrams to complement our documentation. Sarah has a good eye for detail and is a real asset during the testing phases of our projects.

Focusing your skills

Developing a plan

You need to be able to produce a plan with SMART targets to implement a computing technology development in an organisation. Here are some simple tips to help you do this.

- Information gathering is wide ranging and you need to be clear on your sources of information and the techniques you will be using.
- Implementation planning will need you to be very clear on the requirements to identify the project parameters, what the product is intended to achieve and the scope of the implementation.
- Managing risk will involve you planning how to train both the users and support staff with the support mechanisms you would offer them.

Reviewing a plan

You will review a plan to implement a computing technology development in an organisation. Here are some simple tips to help you to do this.

- How will you obtain feedback from the stakeholders and subject experts?
- Review and analyse any ethical impacts on the environment, employment and wider society that your computing technology development might have.
- How will you utilise your skills, knowledge and behaviours, particularly with regard to communication skills and your ability to convey intended meaning with your use of tone and language? You should respond constructively to the contributions of others.
- Did you regularly review your progress against your SMART targets?

Getting Ready for Assessment

Lizzy is working towards a BTEC Level 3 National Extended Certificate in Computing. She was given an assignment with the title 'Understand the impact of developments in computing on an organisation' for learning aim A. She had to write a report with these section headings:

- Hardware and software developments
- Changing markets and new opportunities
- Emerging technologies
- Big data, data warehousing and data mining
- Issues and risks

Lizzy shares her experience below.

How I got started

I set about using several search engines to research the impact of developments in computing on an organisation. I didn't want to just rely on Google™ for this. The websites I thought most useful were bookmarked into a browser folder that I made for this research.

I met up with one of the IT team at my centre to talk through what they knew about emerging technologies. This alerted me to current developments in drone technologies and how a swarm of drones can be used by a fire crew to map out a burning building.

How I brought it all together

I created a Word® document for my report with headings for each of my sections. I also added introductions, a summary and a bibliography section. I typed my name into the header and page numbers into the footer. I always like to work in this way, to get the structure of the document created and then to fill the gaps. This makes it easier not to miss anything and also helps to spread the work out between sections.

The content was just down to reading my research and typing it up. This sounds simple, but I think there's a lot of skill needed here in getting the searches right and in selecting which of my findings to actually use. Sometimes I use double quotes (") in my web searches to keep phrases (rather than the individual words), and I get much better results. I usually look for a date on my findings and am most interested in using results that I can read and fully understand so that I can write them up in my own words. It is important not to use plagiarised work!

What I learned from the experience

The biggest surprise was meeting the IT team member and talking around some real-life computing. Listening to lessons and reading books and searching the web are OK, but there's nothing like the genuine article. I was impressed by the amount of computing knowledge she had and how easily she used technical terms in her conversation. I'm now really looking forward to when my computing career takes off!

Think about it

- Do you have a plan with timings so that you can complete your assignment by the submission date?
- Is there a workplace or organisation that you can visit as part of your research?
- Are there any experts who you can meet to discuss aspects of the work?

Computer Games Development 14

Getting to know your unit

Assessment
You will be assessed by a series of assignments set by your tutor.

The UK games industry is one of the fastest growing job markets today. The UK is a world leader in producing amazing, interactive experiences for desktop PCs, consoles, and handheld and mobile devices. The computer games industry is a team-based and fast-moving area that involves many different job roles. The production of computer games requires an understanding of the whole process from concept art through to formal testing. You may want to become a games programmer, concept artist or 3D animator but, for whichever role you want to focus on in the future, you will have to have a good understanding of the entire games development process first. In this unit, you will investigate how and why people play games, what is changing in games hardware and development, and how to design and create games.

How you will be assessed

This unit will be assessed by a combination of theory, design and practical tasks set by your tutor. Throughout this unit, you will find assessment practice activities that help you work towards your assessment. Completing these activities will not mean that you have achieved a particular grade, but you will have carried out useful research or preparation that will be relevant when it comes to your final assignment.

In order for you to complete the tasks in your assignment successfully, it is important to check that you have met all of the Pass grading criteria. You can do this as you work your way through the assignment.

If you are hoping to gain a Merit or Distinction, you should make sure that you present the information in your assignment in the style that is required by the relevant assessment criteria. For example, Merit criteria require you to analyse and discuss, and Distinction criteria require you to assess and evaluate.

The assignment set by your tutor will consist of a number of tasks designed to meet the criteria in the table. This is likely to consist of a written assignment but may also include activities such as:

- writing an article about the current trends in gaming
- designing a game from a set client brief
- creating and testing a 2D or 3D game level.

Assessment criteria

This table shows what you must do in order to achieve a **Pass**, **Merit** or **Distinction** grade, and where you can find activities to help you.

Pass	Merit	Distinction
Learning aim A Investigate technologies used in computer gaming		
A.P1 Explain social and technological trends of computer games. **Assessment practice 14.1**	**A.M1** Discuss how current and emerging technologies impact on how games are designed and developed to meet the requirements of the users and the larger computer games industry. **Assessment practice 14.1**	**A.D1** Evaluate the impact of current and emerging technologies on the design and development of computer games to meet the requirements of the users and the computer games industry. **Assessment practice 14.1**
A.P2 Explain how current and emerging technologies impact on computer games design and development. **Assessment practice 14.1**		
Learning aim B Design a computer game to meet client requirements		
B.P3 Produce designs for a computer game that meet client requirements. **Assessment practice 14.2**	**B.M2** Justify decisions made, showing how the design will fulfil its purpose and client requirements. **Assessment practice 14.2**	**BC.D2** Evaluate the design and optimised computer game against client requirements. **Assessment practice 14.2**
B.P4 Review the designs with others to identify and inform refinements. **Assessment activity 14.2**		
Learning aim C Develop a computer game to meet client requirements		
C.P5 Produce a computer game to meet client requirements. **Assessment practice 14.2**	**C.M3** Optimise a computer game to meet client requirements. **Assessment practice 14.2**	**BC.D3** Demonstrate individual responsibility, creativity and effective self-management in the design, development and review of a computer game. **Assessment practice 14.2**
C.P6 Test a computer game for functionality, usability, stability and performance. **Assessment practice 14.2**		
C.P7 Review the extent to which the computer game meets client requirements. **Assessment practice 14.2**		

Getting Started

It is important to understand all of the different roles involved in the computer games development workflow. Write down a list of all the different jobs that you think are involved in the making of a computer game, splitting your list into different categories, such as artistic or technical.

A Investigate technologies used in computer gaming

Computer and video games are big business and a continuing source of exciting and creative jobs that require strong IT skills and original ideas. The next big successful title could be just around the corner and one of the most exciting elements of the computer games industry is how small teams with great ideas can become international success stories. Different **genres** can grow out of new titles and create entire new fan bases of dedicated gamers, sometimes forming strong communities of players. Many of the recent game genres have grown out of innovative technologies such as virtual reality, augmented reality, new operating systems and advances in streaming.

Gaming trends and society

The original gaming device was a cathode ray tube amusement device in 1947. There have been nearly seven decades of gaming developments since then and people access and enjoy computer games in many different ways. These different trends have led to an ever-evolving industry that seeks to bring interactive entertainment to all areas of society, whether they are casual gamers wanting a few minutes of distraction with Angry Birds™ or hard-core gamers who eagerly anticipate the newest release of the Halo® series.

Figure 14.1: A scene from a video game

Popular genres

Innovation and originality are keys to the success of a game publisher and players are always willing to explore different genres of games. One of the most popular genres in gaming is the First-Person Shooter (FPS) genre where the player sees through the eyes of a character and must fight through different levels to achieve a particular goal, for example Call of Duty®. Genres can grow and change, they can spawn sub-genres that take the original definition of a genre and spin it off in a different direction, or they may become hybrid genres that take elements of other genres and merge them together. An example of this is an online FPS, such as Star Wars™ Battlefront™, which features many players fighting at the same time. This game has all the features of an FPS game but it also fits into the **massively multiplayer online (MMO) game** genre. So it could be called an MMOFPS. To muddy the waters further, this game also allows you to switch to third-person mode so that you can see the character you are playing.

Key terms

Genre – a category of computer game that describes the style of play, types of challenges and the perspective of the player.

Massively multiplayer online (MMO) game – a game played by multiple players, across the internet, all online at the same time.

Role-playing games (RPGs) have been popular since before games were created digitally. Classic pen and paper games such as Dungeons and Dragons™ saw players work against a dungeon master, having chosen a particular character and, as the game progressed, their character would grow and improve as they made choices. This role-playing concept works very well in computer game format and some RPGs, such as The Elder Scrolls V: SKYRIM®, have had massive success. Other games developers have noticed the appeal of RPG game players improving their characters and have been adding this feature to titles from other genres, such as Call of Duty®, enabling them to 'level up' and receive perks and rewards.

The nature of constant change within computer games, of borrowing of ideas from other genres and adapting to player feedback, means that it can be difficult to write a definitive list of genres but first-person shooter, role playing games, puzzle games and sports games are a good place to start.

Games bring people together as a common interest and popular pastime

Research

How do people play games together? Try to list as many different ways in which traditional games (pen and paper, board games etc) and computer games provide multiplayer activities.

Types of player

As computer game titles can be categorised into different genres, it is also important to consider different types of player. When a game studio pitches an idea for a new title to a publisher, one of the most important factors they have to consider is the audience of the game they want to make. It is impossible to make a game that will please everyone as players have different tastes and interests, and psychologists such as Richard Bartle and David Keirsey have even made academic studies into how and why people play games. Table 14.1 looks at the main factors that determine the types of player.

The table represents basic demographics, that is, measurements used to put people into different categories so that their likes and dislikes can be understood more easily. However, there are other considerations to look at when designing games.

In recent years, as home broadband speeds have increased, online games have soared in popularity and the previous generation of home consoles (PlayStation 3®, Xbox® 360®) were designed to appeal to people who wanted to play together in engaging, online worlds. Our most recent generation of consoles (Wii U™, PlayStation 4® and Xbox One®) have all included the ability to play online and stream content so that other people can watch the play and comment on the skills of the players.

Key terms

Casual gamers – people who only play games for short periods of time and prefer simpler games.

Immersive – a term which refers to how focused you are on the experience that you are having. An immersive game will keep your attention for long periods of time and block out distractions. It should make you enjoy the game more, but only if you have the time to spend on it.

Franchise – a series of game titles that feature the same world, the same characters or the same setting.

Table 14.1: Main factors that determine the different types of computer game players

Age range	In the UK, games are rated by PEGI. PEGI decide what is appropriate for different age groups. Game designers will tend to create bright, cartoon adventures for younger players and darker, more realistic worlds for adult players.
Gender	Despite being commonly considered to be a 'boy thing', gamers are split between male and female, with roughly 60 per cent of players being male and 40 per cent female. Game designers often design children's games for a specific gender, but the older the target audience gets, the less targeted to a particular gender the content becomes.
Time commitment	Possibly one of the most important factors is how much time a person is willing to spend playing a game. This can affect the design of a game greatly. **Casual gamers** will happily load a game for a few minutes at a time. They may play on mobile devices while travelling or download digital games that cost a lot less than boxed games. Casual gamers will not be particularly loyal to one genre, worried about saved games or necessarily interested in sequels.
	Players who want a much more **immersive** experience will spend hours exploring their favourite worlds. They will be loyal to particular brands and spend a lot of their spare time playing games. Immersive gamers may also be referred to as 'hard-core gamers'. Some people may criticise the time they spend on games, but is it any different from spending hours watching TV?
Theme choice	The content or style of a game and any ideas that tie all of its features together are called its theme. People may choose a game based on its story, setting or design. There are many different game themes: a fantasy adventure, a realistic war game, a puzzle game with fun characters or a game that follows the stories and characters from a well-known film or book **franchise**.

Game production

The production of a game is a complicated event. It usually begins with an idea being pitched by a **game development studio** to a **game publisher**. This pitch will contain an overall concept of the game with details about the characters, **game mechanics** and who the game is being designed for, that is, the audience. Without an identified audience, there would be no financial incentive for making the game. Sometimes the publisher will approach the studio with an identified audience or game genre and/or theme and ask them to create a game for that market. After the studio has been given the funding to start development, they will begin by creating **concept art** that illustrates the graphical style and theme of the game (see Figure 14.2).

Key terms

Game development studio – a team of people who create computer games.

Game publisher – a company that releases games to shops or online platforms, and pays for development.

Game mechanics – the way a game world works, its features and its rules such as double jumps or collecting items.

Concept art – drawings and paintings created before a game is developed to show how the game world should look and feel.

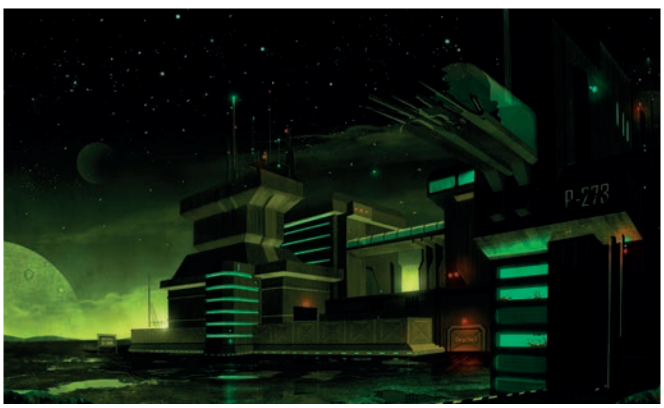


Figure 14.2: Concept art

The concept art will be referenced throughout the game's development so that everyone on the development team knows what the game should look like, and the kind of feelings that the game should be evoking in the player.

Mainstream publishers

Some publishers are large international companies, known as mainstream publishers, such as Electronic Arts (EA™ Games) or Nintendo®. These mainstream publishers are responsible for funding the production of many different game titles at the same time. Smaller publishers, such as Telltale Games®, will usually specialise in one particular genre and will often release only one game at a time.

Indie games

When development of a game is started before it has a publisher, it is known as an indie game. Studios that make indie games fund the development themselves. This means that they do not have the same restraints placed on them by publishers during development so they can make their own decisions about the content and style of the game. This often leads to innovative and original game designs but it also means that the studios can struggle to pay the bills, as they do not make any money until the game has a publisher. When a studio has proved that there is a market for their game, through research or testing, a publisher will fund their project. Good examples of successful indie games are Minecraft™, Super Meat Boy™ and No Man's Sky™.

Indie games are able to start making a profit when they find a publisher. Often the profit is fed back into the development of their next game, before it gets a publisher. Indie games usually get published as digital downloads because getting a game disc printed, boxed and shipped to game shops all around the world costs a lot of money. It is partly because there are so many good indie games now that digital download providers (such as Steam™, PlayStation® Store) have become so popular. Mainstream publishers will now decide before development if a game is going to be printed on disk or released through a digital platform. If they expect to sell to a wide audience then they will make the investment in a boxed game sold in a shop, but if the game is more niche then it may only be released digitally, as there is a greater financial risk in printing on disc because it costs more.

Crowdfunding

Indie developers are able to raise funds directly from players through crowdfunding websites such as Kickstarter™ or Indiegogo™.

Crowdfunding is a recent phenomenon and it has really exploded over the last ten years to a point where it has gone from being a gimmick, which could be used to top up the finances of a struggling studio, to a viable means of funding large-scale projects. Virtual reality hardware such as the Oculus Rift™ would not exist without crowdfunding. Equally, some very successful games have been funded through this model, for example Broken Age™, which was one of the first games to raise far more funding than it asked for. Star Citizen™ is the most successful crowdfunded game to date, having made over £25 million.

The crowdfunding model differs from traditional game development not only in how the money is raised but also in the inclusion of 'stretch goals'. This means that developers will commit to a certain scope for the game, but given additional funding they would make additional promises. These 'stretch goals' may be different platforms for release, extra levels or a virtual reality mode. One of the disadvantages of crowdfunded games is the delays to titles, over which customers have no control, as extra features can push back delivery dates and people can often wait a long time for their games.

Free-to-play

Another model of game production is the free-to-play model, which has become a mainstay of mobile apps games development. Using this approach, smaller studios and publishers release games for free and then include a number of paid-for upgrades or features. The idea is that players will get hooked on the game when they can play it for free and will then be willing to spend money unlocking extras. These free-to-play games are also known as freemium titles. Some players find it frustrating to begin a game that they think is free only to find later that they have to pay to keep playing. Publishers of free-to-play games can avoid making players pay for the games after a certain period by placing adverts in the games, which pay for the cost of development.

Link

The development of mobile games apps is covered in *Unit 17: Mobile Apps Development*.

Artificial intelligence

Artificial intelligence (AI) is the name given to the programming that makes machines (or non-playable characters (NPCs), in the context of games) seem like they are thinking for themselves.

Programmers have to think about how an NPC in a game should react to where they are or what they are supposed to be doing. A soldier guarding a gate should be looking in certain directions but should have moments where their attention lapses so that the player can sneak past. All of this character behaviour has to be coded into the game.

Modern game AI has developed to the point where enemies are able to have realistic reactions to players: such as an enemy blocking an attack in a fighting game, or more strategic behaviour such as enemies responding to patterns in the player's choices by avoiding a 'duck and cover' attack or trying to flank enemies, in multiplayer games. The more sophisticated the AI, the more complicated the programming required to create it.

Search algorithms

Different **algorithms** are used when designing AI and programmers often use search algorithms, such as the A* Algorithm, which is a path-finding algorithm. A search algorithm allows the AI attached to an NPC to solve problems, such as the problem of where to go next. The problem of deciding the best route to take from one point to another is quite easy for humans: we look at the area around us and decide the best way to get somewhere using the smallest amount of effort. It is not as easy for an NPC (often an enemy) in a game; the NPC has no real concept of where it is or where it should be. Therefore, when a player is seen by the enemy at the end of a hallway that contains several crates, barrels and piles of rubble, it needs an algorithm to figure out how to get down the hallway to the player. The A* Algorithm would break the hallway and obstacles down into a series of steps and work out the best set of steps to get the NPC from its starting point to its goal. There would be a choice of different routes to take and the algorithm uses mathematical optimisation to work out the best route to take.

Key term

Algorithm – a set of instructions that are executed in order to solve different computer problems.

Mathematical optimisation

Mathematical optimisation is the selection of the best solution to a problem when given a selection to choose from. It forms a part of all AI and is often tied to a game's difficulty setting. Most game NPCs would have the ability to get rid of a player quickly, if programmed correctly, but that would render the game impossible. Therefore, a game's difficulty setting will change the possible solutions available to NPCs, from a simple approach such as lowering the enemy's health value or making a less powerful adversary, to more sophisticated AI such as the enemy being able to change tactics.

Logic

An important factor in creating realistic AI is considering the logic and patterns of behaviour that a human would normally follow. Does the enemy behave in a way that is logical and reasonable? If it emerges in the testing phase of a game's development that the answer is no, then the AI programmers must go back to their code and improve it until a player is not puzzled by the behaviour of an enemy. This is a difficult task as games are very complex, interactive software applications and it is hard to predict what players might do in a game. In fact, players often cheat by finding an 'exploit' or vulnerability in a game's AI. This often leads to developers having to find a fix that they then patch onto the game as a download after the game has been released.

Emerging technology

One of the things that makes the computer games industry so exciting is the rapid pace of development and new ideas that change our game playing experiences. New ideas are driven by improvements and innovations in the hardware and software that we use to play games.

Research

What improvements to games hardware have been made in the past few years? Have a look at the technical specifications and features listed by different hardware manufacturers and see if you can come up with a list of the technological features that define the current generation of games platforms.

Virtual reality

One of the fastest growing areas is virtual reality (VR). Back in the 1990s, VR was set to be the next big thing but the headsets were very expensive and the processing power was too slow, so the technology never took off. However, more recently crowdfunding has brought about the first version of the Oculus Rift™ headset, which has proved to be an incredible success and spawned imitators from Valve™ and Sony®.

Figure 14.3: Virtual reality is creating a whole new way of engaging with games

VR in its current form consists of two screens, one over each eye, with special lenses that magnify the screens so that they fill your field of vision (see Figure 14.3). Added to the headsets are accelerometers, which measure which direction you are looking in, and gyroscopes, that measure how much you have turned your head. Often a positional tracker is paired with the headset, which follows your head's position in space so that you can move forwards and backwards. This positional tracker can be combined with motion controllers, and even a treadmill, to create total movement in a virtual world.

Augmented reality and wearable technology

Microsoft® have taken a different approach to the new headset phenomenon and have focused on augmented reality (AR) instead of VR. AR uses cameras to capture the real world and layers virtual game assets over the top, so that they appear to exist in the real world (see Figure 14.4).

AR already existed on various mobile apps and in game form on the Nintendo® 3DS™ and PlayStation Vita® but Microsoft®'s HoloLens™ seeks to outdo them all by creating engaging experiences with virtual characters in the real world.

Once you have exhausted all the games on your PC, console, mobile and handheld games devices, you can look to smart watches to top up your gaming urges (see Figure 14.5). The Apple® and Android™ smart watch app stores both have a growing range of simple games, which you can play on your wrist. These games are restricted by the size of the screen that they are played on, but sometimes a physical restriction can be the catalyst for an amazing game idea. Lifeline™ is an example of an Apple® Watch® game, where the player receives messages from an astronaut stranded on an alien moon. The game's interface is simple text but it is an engaging and exciting story with delays built in so that players are not staring at their watches for hours on end.

Figure 14.4: Whilst virtual reality places you in a different world, augmented reality brings virtual objects into our world

Figure 14.5: The Apple® Watch® gives you access to apps on your wrist

Digital distribution

It is not just headsets that are changing the way we play games. Digital distribution platforms such as Steam™, GOG™ and PlayStation® Store have changed how we buy games. One of the most dominant games providers, Steam™ by Valve™ is also one of the best places to get indie games, which can often cost a lot less than console disc-based titles. Disc-based games are often priced the same as their digital equivalents on the console's own digital stores, as there are not any incentives for console publishers to discourage their audience from buying games in a traditional shop, because if the shops cease to exist then there are fewer places to buy the console.

Steam™ has created its own operating system (OS). Steam OS is a free, **Linux**-based OS that prioritises gameplay and is available on 'living room' PCs. 'Living room' PCs are compact PCs that are connected to a TV and are designed to be used in a similar way to a traditional games console, by using the digital platform as the main source of content.

Key term

Linux – an operating system (OS) which is released 'Open Source', meaning that the source code that creates it can be downloaded and adapted by anyone. Due to its adaptability, it is very popular and comes in lots of different versions such as Steam™ OS.

Streaming

Streaming has become a huge part of games culture with websites such as Twitch™ and YouTube making gameplay videos highly popular because they enable people with limited budgets the opportunity to preview a game that they might be interested in buying. Many people are used to streaming music and films to different devices, and games can be streamed too, using services such as PlayStation Now®. Subscribers can rent games for set periods and download them to their device, be it a console, PC or a smart TV.

Discussion

Which of these emerging technologies would engage you as a gamer? Discuss, in a group, who you think these emerging technologies would engage and whether or not new audiences for games may grow out of these emerging technologies.

Security of integrated services and multiplayer environments

As game systems become more reliant on digital services and streaming content, there is an increased requirement for users to share their personal data with the services that they are using. Most of the digital platforms require stored credit card details and they will often ask customers to connect their social media sites so that customers can share details of what games achievements they have gained or levels they have completed. This presents a big security risk, as the information could be used by criminals to steal a customer's identity. For example, Sony® was the victim of a series of cyber attacks in 2014. All the companies offering games platforms have to be very careful that new updates and upgrades to their software do not inadvertently create a breach that criminals can exploit.

MMO game providers can also be the targets of hackers and they often have less security in place than an online bank or shop. It is important that they protect their users' data because, despite being a game rather than a bank, they are still about customers. Players may also have their game content stolen and sold by hackers.

Players do not just have to worry about the security systems in place on MMO games, but they also have to be careful about the information that they give out if they are having conversations with strangers in the game through text or voice chat. Criminals are able to get a lot of information out of players without them realising it, by pretending to be friendly. Before they know it, the player's password has been guessed and they are locked out of their own account. (This could be their login for an MMO such as World of Warcraft® or a digital distribution account such as Steam™ or Google Play™.)

Gaming technology

Keeping up to date with games technology, and how it is changing, is one of the main challenges for anyone involved in the computer games industry.

Benefits and limitations of different platforms

Players face the added challenge of choosing which platform to invest in. While developers may often want to create games for the most popular platforms, each one comes with its own benefits and limitations. This can be seen in Table 14.2.

Hardware

Most gaming hardware is made of the same core components to allow digital information to be displayed on screen and interacted with by the player.

Central processing unit

The central processing unit (CPU) is the brains of the computer. It carries out all of the instructions sent to it by a computer program using thousands of tiny switches to perform arithmetic, logic and input/output. In the case of a game program, the CPU is responsible for working out all of the game's system requirements. Most modern PCs use an Intel® or AMD CPU and the performance of PC games (such as that of Steam™ titles) is dependent on the power of the processor. For example, The Witcher® 3 is a huge open world RPG with very good graphics and its minimum system requirements include an Intel® Core™ i5 processor that runs at a speed of 3.3GHz (a fast processor which is not sold cheaply), but its recommended requirement is an even more expensive Intel® i7 processor running at

Table 14.2: Benefits and limitations of different platforms for games developers

Platform	Type	Benefits	Limitations
Windows® operating system PC	Personal computer or laptop (desktop)	Easy games development due to widespread availability of PCs; broad user base; access to Steam™	No standardised technical specifications; some games will not run on lower spec PCs
Mac®	Personal computer or laptop (desktop)	Works with most digital platforms; powerful systems with fast CPUs	Fewer games are released for Mac®
PlayStation 4® and Xbox One®	Console	Hugely popular consoles; strong hardware; OSs dedicated to playing games	Can be expensive to develop for; getting disc distribution requires working with large publishers
Nintendo® Wii U™	Console	Large fan base but not many titles compared to other platforms so developers have an opportunity to take advantage of this!	Very specific tablet-based controls mean that games cannot be ported onto other systems
Apple® iOS devices (phones and tablets)	Mobile devices	Hugely popular with owners willing to invest in higher priced games than non-iOS® mobile apps	All products have to be approved by Apple® who can have quite strict standards; development usually restricted to Mac® unless using a games engine
Android™ devices (phones, tablets and notebooks)	Mobile devices	Much cheaper development costs than Apple®; open source system	So many different types of devices means quality testing can be difficult
Adobe® Flash®	Web-based platforms	Easy to animate and create great visuals on	Flash® does not have simulated physics; due to security concerns, many browsers are discontinuing their Flash® support
HTML5	Web-based platforms	Supported on all web browsers, desktops and mobile devices; no plugins necessary	No built-in support for 3D, gamepad or to save games

3.4Ghz. Given this, many PC gamers will overclock their CPUs. 'Overclocking' means to force the CPU to run faster than its recommended manufacturer speed and, while it can be done with stability by changing the voltage, this will invalidate the warranty and put your PC at risk.

Games consoles are able to use slower CPUs because they optimise their games to run with fewer details. For instance, they may use less real-time shadows or lower-resolution textures. They also do not have to worry about running large operating systems in the background, like Windows® or OS X®.

Mobile devices, such as smart phones, run with much slower CPUs, which is why they are unable to run large-scale games with complex graphics.

Graphics processing unit

A graphics processing unit (GPU) works alongside the CPU and its sole responsibility is to manage the production of images on the display. The more complex the graphics, the more work for the GPU. When playing PC games, the GPU can be part of the main **motherboard** but is more commonly a separate card connected to the motherboard through an expansion slot. Most GPUs have their own separate random access memory (RAM) (see below) and PC games will often specify the minimum power of the GPU needed by a game to run. Consoles also have their own GPUs, as do mobile devices.

Key term

Motherboard – a circuit board within a PC that connects all the main components.

Memory and storage

Memory is split into two different types: read-only memory (ROM) and random access memory (RAM). ROM is used in games as storage for the game's software. This is usually a DVD or Blu-Ray disc (downloaded games from Steam or mobile app stores are stored on a hard drive or flash memory) but cartridges were used in the past in consoles and handheld devices. RAM storage is used to temporarily store data when the processor is working on it, and the more RAM available, the faster things usually run. Consoles and mobile devices use the amount of RAM they have as a selling point of their systems whereas on a PC it is a lot cheaper and easier to upgrade the amount of available RAM.

Long-term writable storage, unlike ROM that is read only, usually comes in the form of a hard drive or flash memory that will store not only the game data itself but also user-based data such as game saves and leaderboards. As games get more popular and people want to change devices when new models come out, cloud storage for game data has become a staple feature of consoles. Letting players store game data and game saves in the cloud means that they are able to continue playing from the same point if they upgrade or change their machine.

Output

Sound and display graphics are major elements of games and, as such, form the biggest share of the output. Gaming PCs and consoles use TVs or monitors to display the graphics but some, like the Wii U™, will have secondary screens for displaying additional information or providing a screen to use when someone else wants to watch TV. Smartphones and tablets are currently in a hardware war to outdo each other's display capabilities, with 1080 pixel screens becoming the standard and 4K screens providing unbelievable detail on a device that you can carry around in your pocket.

Sound is important, as it is one of the main ways in which the game communicates with the player without distracting them with text. A positive sound effect will inform the player that something good has happened and part of a game's learning curve is to remember which sound effects mean which benefits or punishments.

Another output of a game could be haptic feedback, which is a physical sensation such as a tremble or vibration, which is usually provided by controllers (see Figure 14.6).

Due to copyright we are unable to reproduce this content here

Figure 14.6: Game controllers can provide a rumble feature which provides extra feedback about events in a game

Input

The way in which a player inputs information into a game is more crucial than any other aspect of software for computer games. The slightest delay in the interpretation of input by a game could mean the difference between life and death for your character.

Many PC players will argue that keyboard and mouse controls are better than those of a gamepad controller, but it really comes down to personal preference. Keyboard and mouse controls offer more customisability but players can use gamepad controllers when sitting on a sofa rather than at a desk.

Recent developments in voice and kinetic controls have created new and innovative ways to control games that not only challenge designers to come up with new ways to input instructions into games, but also open gaming up to people with physical disabilities who may struggle to hold traditional gamepads.

Touch input has become a big consideration as games for mobile devices have become more commonplace, because users will need to use the touchpad of their smartphone or tablet to play games. Some gamepads now include touchpads so that casual gamers are tempted to move across to console titles.

Connectivity

As we move into a time where society is always online, gaming devices are required to stay connected to the internet as much as possible. This allows operating systems to perform fixes and updates and for games to download patches to deal with bugs and exploits. PCs and consoles maintain their connections through local area networks (LAN) or wireless connections, while mobile devices can also enjoy wireless through mobile data network connections.

Link

For more on new hardware technologies, see the Emerging technologies section.

Software

Games are software applications, written using a programming language, and run on an operating system. They are a lot more complicated than most software applications but they are written using many of the same programming processes and techniques.

Operating systems

PC games are run on either a Windows® operating system, Mac® OS X® or Linux operating system. These operating systems differ in popularity and this is the main deciding factor in how many games are made for these systems. While most family homes will have run a Windows® operating system PC, Linux is a more specialist operating system and Apple® Mac®s are traditionally used by designers. If the priority use of the operating system is not gaming, then fewer games will be made using that operating system. However, digital platforms have started to change this.

PC operating systems are made to work on a number of different hardware configurations (on various different PCs, laptops and notebooks), unlike console operating systems which are designed with far fewer features and for one specific piece of hardware (one version of a console, eg Xbox One®). Console operating systems are known as proprietary systems, which means that the hardware manufacturers create them, and the source code is kept private and is not intended for use anywhere else. Most mobile devices run on either Android™ or Apple® iOS, but some run on Windows® operating system. A newer operating system is Steam OS by Valve™ which has been designed for PCs that are primarily being used for gaming.

Programming languages and graphics options

The programming language in which the game is written can also vary quite a lot depending on the platform that it is intended for. The dominant language for developing games is C++, which is an **object-oriented language** that has been popular for nearly thirty years. Other languages that use a similar structure are C# and JavaScript. C# is used within games engines and JavaScript is used to create Android™ games. Different programming languages tend to be tied to different platforms. Some languages are described as 'light weight' and will only run within a games engine, such as JavaScript. These 'light weight' scripting languages rely on existing assets and features to keep the amount of code low as a result. Developers using Mac OS may use Swift to create game code.

Key term

Object-oriented language – uses code that is organised into objects, which can be used to make it run in a fast and robust manner.

Graphics software also needs to be considered. Systems use application programming interfaces (APIs) for managing the tasks related to the software and the GPU. The APIs are largely sets of routines and protocols for making the development of games easier. DirectX® is a Microsoft® graphics API and OpenGL® is the open source alternative.

Device drivers

Device drivers are another important type of software used for running games. The drivers are responsible for identifying the hardware devices that have been connected to the operating systems. Drivers tell the PC, console or mobile devices which hardware is trying to communicate with it and how it should work. This is how peripheral devices connect to games, such as newer controllers, microphones or even dance mats. Console manufacturers such as Nintendo® or Sony® will often release extra hardware devices in order to prolong the lifespan of the system and keep players interested.

Audio options

Audio is another important area of software. The music and sound effects are a massive part of the feeling and ambience that designers want to create for a game and one of the best ways to evoke feeling in a player is through music. A game will have different music in the different levels and these will be similarly themed to generate a common link across the game. Sound effects are used to teach players when they have done something positive or negative. Music files can be quite large and developers have created a number of different file formats in order to make the games run more smoothly, especially for online games. These file types are WAV, MP3, FLAC and AAC.

PAUSE POINT Create a technical specification for both a photorealistic 3D, online FPS game and one for a 2D casual web game. What are the main differences and what decisions about software do you have to make?

Hint Think about the components that would be required: the basic requirements are a CPU, GPU and memory, but what extra hardware would make the gaming experience more complete?

Extend Can you find examples of games that stretch the capabilities of their platforms? Are there any 3D games for the web? If so, what sacrifices do they seem to make in order to run smoothly?

Games engines

Writing a computer game from scratch is a large undertaking and can involve writing thousands of lines of code. One way to make this easier is to use a games engine. A games engine is a piece of software, which is designed to make games. Most games engines are a combination of designer and programming environments in which the user is able to place all of the graphical assets for a level and then write the code that makes it interactive. There are a number of popular games engines in use today. Some of these are proprietary, meaning that they are only available to particular publishers or the large games studios that own them. However, some others are available to anyone, either for free or for a subscription fee. Popular games engines for multiple platforms are Unreal Engine 4 or Unity® software (see Figure 14.7), which are able to build games for PC, consoles, the web or mobile devices. Other engines are designed for one particular platform, for example XCode® is designed for Apple®'s iOS.

Due to copyright we are unable to reproduce this content here

Figure 14.7: The Unity® software is a powerful application used to build games quickly and efficiently

Rendering engines

One of the main jobs of a game's engine is **rendering**. This is usually done through one or more virtual cameras that point at the actions within the game that the players need to be focused on. The speed at which the game is rendered is measured in frames per second (FPS). The faster the FPS, the smoother the game appears to the player. When a gaming system is not powerful enough to run the game that the player has loaded, the FPS is usually the first casualty and a game may appear to run slowly or lag.

Physics engines

As all computer game worlds are virtual, all of the events and rules that occur within them have to be programmed. Some of the hardest elements to create involve the laws of **physics** that need to be applied to the world. Most games engines include built-in physics that takes the responsibility away from the programmers and allows for faster game production. Other engines that do not include built-in physics can use APIs to add physics. An example of this is the Box2D physics engine that was used to create Angry Birds™, one of the most popular physics-based puzzle games ever made. Another physics engine is the Havoc® physics engine, which has been used in many popular 3D games such as Assassin's Creed®.

Key terms

Rendering – the process of converting game assets and environments into 2D images that can be displayed on a screen.

Physics – in the real world, gravity, mass and other laws of physics apply naturally but, in a world created by a computer programmer, the laws of physics have to be made to apply through coding the game correctly.

Collision detection – the process of checking to see which game objects have collided with each other.

Collision detection

Once the objects in a game world have physics, they have the ability to collide with one another. Working out the correct way for objects to collide which each other, and what occurs afterwards, is known as **collision detection** and is another important role of a games engine. Objects in a game world may have solid colliders surrounding them so that game characters are not able walk through them. These colliders may be the sides of a building, the floor or even invisible walls to stop a player falling off the edge of the game world. The player's character will also be surrounded by a collider that detects where their boundary is and stops them from moving through walls. A game character, whether it is 2D or 3D, has a very complex shape and it would be too taxing on a system to make every part of their body a collider. Instead, a capsule-shaped collider is often placed around them (see Figure 14.8) which is why, if you are careful, you can sometimes see a character's arm or leg moving through a solid wall in a game. The more you learn about how games are made the easier it is to spot things like this when you are playing them.

Scripting

A games engine will always provide the ability to add code in order to make things happen in the virtual world that has been created. Code is added using a **scripting language** such as JavaScript, or a full programming language like C++. The Unreal Engine contains a visual scripting language called Blueprints. This is an easy way to create interaction without having to write lines of code.

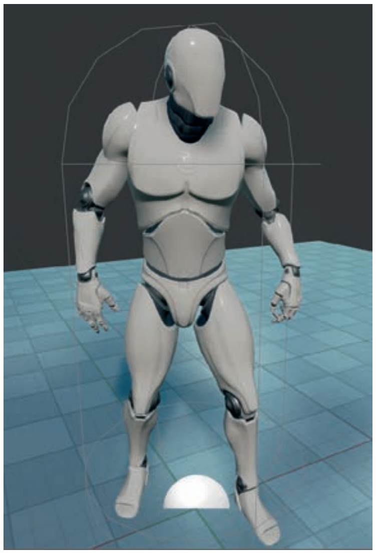


Figure 14.8: Character surrounded by capsule-shaped collider in Unreal Engine to enable collision detection

Graphical nodes are used to create movement, interaction and game mechanics like health or ammunition. Blueprints is similar to Scratch, a visual scripting language used to teach programming, but Blueprints provides a lot more functionality. The downside of using a visual scripting language is that the more complex the game's mechanics, the larger the number of nodes on screen, and it can soon look like a scramble of lines and nodes. If this happens, it would be better to switch to C++ and write the game in traditional code.

Key terms

Scripting language – a programming language that requires a separate application to run, such as a games engine or web browser.

Graphical nodes – colourful blocks used to represent the different functions of a programming language used to add easy interaction.

Animation

The final responsibility of the games engine is animation. Animation involves making different elements in the game move. There are two types of animation: vertex-based animation (which moves parts of a 3D mesh or 2D sprite) and skeletal animation (which uses a rigged character and sets of different animation cycles). An example of vertex-based animation would be a door opening when the player approaches it or a floating coin spinning around.

The image below (Figure 14.9) illustrates skeletal animation in the Persona system within Unreal Engine. A character created in 3D can be rigged with a set of virtual bones that are, in turn, attached to different parts of the mesh. When a virtual bone is moved or rotated, the mesh connected to it will also move. The player will never see the bones but the engine knows that they are there and understands how they are supposed to act. The 3D character modeller will rig the character in a design software application such as Autodesk® 3D Studio Max, weight the bones so that the mesh is connected properly and then create different animation cycles such as a walk cycle, run cycle, jump cycle etc. These cycles are a single animation that ends at the same position as it starts so that it can be repeated for as long as the game requires the character to do that action. For example, when a player is pushing forward on the gamepad, the game will repeat the run cycle for as long as the player pushes forward. When a character is imported into the games engine, the animation cycles will be imported too.

Figure 14.9: Skeletal animation in the Persona system within Unreal Engine

PAUSE POINT Games development can be an expensive process but much of the software mentioned so far is free for learners to use or does not cost anything until a game is published and has made a certain amount of money. Can you find out what the costs are for the software discussed so far?

Hint The websites for each of the different software are the best places to start.

Extend We have discussed game engines, 3D modelling packages and operating systems, but graphic design software plays a large part in games development too. What additional software might you need?

Assessment practice 14.1

A.P1 A.P2 A.M1 A.D1

A mainstream games publisher is interested in creating a new game franchise that will have different titles that embrace the latest developments in current games technology. They want to show that they understand the needs of gamers and the different devices on which they can play. They also want to be seen as being part of the next phase in gaming. They have asked you to make a presentation to one of their development studios to help them understand the opportunities that are available.

They want a presentation which:

- explains the current social trends of computer games:
  - What are the different types of players?
  - Who buys what games?
  - How have habits changed over time?
- explains the technological trends of computer games:
  - What emerging technologies are being developed in the computer games industry?
  - How are existing technologies being used alongside new ones?
- discusses how current and emerging technologies impact on how games are designed and developed to meet the requirements of the users and the larger computer games industry:
  - How is this changing the games that people buy?
  - How have game designs changed to meet them?
  - Who is investing in this technology and what will they gain?
- evaluates the impact of current and emerging technologies on the design and development of computer games to meet the requirements of the users and the computer games industry.
  - What technologies have failed in the past?
  - Are people playing games differently?

Plan

- What am I being asked to do?
- What information do the designers need?

Do

- Am I getting the most up to date information?
- Can I look at how past technology has impacted on game design?

Review

- I can explain what parts of researching the presentation were the hardest.
- I realise where there are still areas of the industry where I have knowledge gaps.

B Design a computer game to meet client requirements

Designing a computer game is a complex process involving team members with many different specialist skills. The design phase is vitally important and rushing this stage can lead to many problems for the development team, or even the players further down the line. However, game development studios are not always in charge of their own deadlines, for example if they are working on a commission for a large publisher. Often they will be given the contract on the understanding that they meet very specific deadlines, so an efficient but also effective design process is crucial.

Computer games design processes and techniques

Understanding the formal methods used to design a game is very important. Each member of the team is trained in their own specialism (art, asset creation or programming) but they all have to understand all the steps in the game design workflow. The first stage of the game design process is the creation of a high-concept design document that will outline all of the game's unique features, storyline, characters and mechanics. This document is then expanded into a game design document (GDD), which covers all of the detailed design specifications for the entire game including how it will be made, how many levels there will be, how the game world works etc. The GDD is then shared between every member of the team and extra documents are written, for example a style guide and a technical specification. These two documents contain even more detailed designs for the specialist teams who will be working on the art or the programming sides of the game.

Mathematical techniques and processes

Maths and logic are a core element of programming and as such they form a crucial part of games design. The technical specification outlines all of the code that is required for the game, the platforms and languages that are being used, as well as details of the developers who will be responsible for creating them. The processes required for creating the functionality of the game and the maths that will make it happen must be considered thoroughly.

Calculations

Calculations are present in all software. Every time we click a mouse button or a gamepad trigger there are lightning fast calculations being performed in order to figure out what should happen as a result. Games use maths in a variety of different ways. The mathematics of graphics is one of the most common uses of this as geometry (the area of mathematics that deals with shape, size and space) is used to position 2D and 3D assets in a virtual world and then manage how they move, rotate and interact with one another. Geometry deals with points, lines, planes and solids and this transfers directly to the 3D modelling of objects where we create vertices, edges, faces and meshes. Games will use geometric calculations as part of their game mechanics. So, if we are playing an FPS where a player must fire a bullet at a target, geometry is used to calculate what happens if the player shoots at a particular angle and from a particular distance. Where would the bullet hit the target if fired in that direction, at that angle and from that distance? Geometry is used to establish which shots would achieve the desired result and which shots would miss. In geometry, the point where a line cuts through a plane is called the intersection, and it would be this value that decides whether the player has been successful or not.

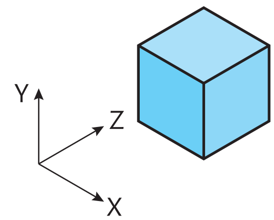


Figure 14.10: All 3D game objects have a position in 3D space which is measured on an XYZ co-ordinate

2D and 3D space

The various different points in game space that need to be monitored, such as the location of a power-up or the exit to the level, will be stored as a vector. A vector is a mathematical way of representing the point in terms of its location on the *x*, *y* and *z* coordinate axes. Based on its distance from the game world's centre point (0,0,0), you can consider the vector value to be the object's address in the game world. Vectors are written as (*x*,*y*,*z*), always in that order, so that all developers and functions within the game understand the vector's location.

Vectors have two main abilities: they can move (or transform) from one place to another or they can be rotated into a new position. The way in which a vector rotates is decided by its pivot point. A spinning coin may have a central pivot whereas a door would pivot from one side. Vector values are also used to describe a direction that something is going to move in, so, if a platform was going to move to (0,0,10), it would move 10 units in the *z* direction but stay in the same position on the *x*- and *y*-axes.

You already know that maths is used in games, for example path-finding algorithms such as the A* algorithm, but it is also used as part of a game's physics. While all objects will have a vector to store their position, some will be marked as physics objects and they will have another vector for velocity, acceleration and mass. These vectors decide the direction in which an object is moving, how quickly it is moving, whether it is speeding up or slowing down and its mass.

Visual styles, graphics processing and editing techniques

The GDD will set out all of a game's design details in terms of how the game should look, and a large part of the design process is dedicated to this.

Worked example: Gravity

Now have a look at how a game would use maths to add gravity to a falling object. Most games will have a game loop function that runs continuously. Every time the game changes frame, the game loop function will run. Some engines call this an 'Update function' or an 'EventTick'.

Now set a numerical value for the gravitational field strength. Both Unreal Engine and Unity® software game engines use a value of about 9.8 m/s^2 (metres per second per second), so the velocity of the object increases by 9.8 m/s every second in free fall. This is the same as the gravitational field strength on Earth.

Setting the gravitational field strength value to 9.8 and giving an object physics means that in every game loop we would say:

If the object is still falling, keep on increasing the velocity by the gravitational field strength.

Which would mean, while the object has not hit the ground, make it fall faster until it does.

Can you see a fault in this approach so far?

The game loop function loops every time the game changes frame and gravity causes an acceleration of 9.8 m/s each second. This means that the game's gravity is going to be stronger than standard Earth gravity because it is going to increase the falling velocity more often than 9.8 m/s each second. The game's programmers would have to calculate the game's likely frames per second and work out a smaller amount of gravity to add with each game update, or only add 9.8 m/s to the falling velocity after one second has passed.

Visual styles

While there are still a few text-based adventure fans out there, the majority of games are incredibly visual products and their individual art styles and design can often be one of the main selling points for the game. For example Limbo™, by Playdead™, studios is a monochromatic game (black and white) with soft lighting and all the characters are silhouettes (see Figure 14.11). This creates an amazing atmosphere of isolation and tension when playing the game and the art style is so important that every person working on the game had to consider it at all times. This means that the style guide for the game would have been of vital importance, especially to the asset designers.

Due to copyright we are unable to reproduce this content here

Figure 14.11: A screenshot from the Playdead™ game, Limbo™, showing its unusual visual style

Graphics processing

Objects in a game will have a texture or sprite applied to them via a shader. Also known as a material, the shader is a piece of code that defines how the object should be rendered graphically. It will set out properties such as the object's colour, texture, reflections, metallic value or outlines. Shaders are very important for maintaining a consistent graphical style across an entire game. They come into effect during the graphics processing stage of designing a game, before all the objects are rendered into 2D images that are then animated onto the screen, because the shaders decide how the objects should look.

Due to copyright we are unable to reproduce this content here

Figure 14.12: Different effects can be created using shaders to allow the game developers to achieve the world designed by the concept artists. Screenshot provided by Ross Everson

Editing techniques

There are many different visual or art styles used in games and this is a major part of what makes games so popular to different audiences. Players may like **cel shaded** games such as Borderlands™, an abstract art style such as The Unfinished Swan™ or a paper craft style such as Tearaway™.

Key term

Cel shaded – is a type of non-photorealistic rendering. It is designed to make 3D graphics appear to be flat. It is often used to mimic the style of comic books.

The art style of the game could be reliant on particular editing techniques. Editing techniques are the tools and features of graphic design software used to manipulate images to get certain effects. This would be the responsibility of texture artists, who may employ a particular technique, such as using a cartoon art style, to create the textures for objects in the game. These textures are created in a professional image-editing software application such as Adobe® Photoshop®.

Platform

Given that platforms vary in terms of capabilities and features, the target platform for a game is a big consideration for the designers. If a game title is intended for one particular platform, then it must be designed to take full advantage of that platform. For example, PC games should take advantage of keyboard and mouse controls as they are the most configurable controllers available and players can decide exactly where they want to map action buttons. Equally, games for the Wii U™ should take advantage of its tablet controller, which has its own touch-screen display that can be used to display extra information and provide additional interaction with the game.

Other titles are designed to be played on multiple platforms. If a game is designed for smartphones, will it work on Apple® iOS, Android™ and Windows® phones? If it does, will it connect to all of a player's game centre social networks and allow players to compare scores or challenge each other? Will it scale up to tablets such as the iPad® or the Nexus® 9? All of these questions have to be answered during the design phase of a game's production. Crowdfunded games may have stretch goals to expand their titles to different platforms after initial design, while games that are funded by mainstream publishers will have set agreements with different platforms before the game is made and they will only expand a game to new platforms if it has done well in sales.

Delivery

Games are designed for either physical or digital delivery, that is, their distribution to customers will either be by means of a physical product (a disc) or digitally (via a download or online streaming). If it is physical delivery, then there will be set deadlines for when the game code needs to be supplied to the factory responsible for printing the discs and game boxes. There will also be an additional amount of time allocated to distributing the game to shops around the world in time for release.

Games that are delivered digitally through platforms such as Steam™ or PlayStation® Store will be given a deadline for digital distribution but it will not require as many steps to make it ready for the player. The choice of delivery method will often depend on the size of the game and the platform for which it is intended. Mobile games must have digital delivery, as smartphones and tablets do not have disc drives. There is the option for both on PC and consoles. Digital games are usually smaller. However, increasingly gamers like to have the choice of delivery method: a hard copy on disc that they can keep or the convenience of downloading their next game.

Game assets

A game asset is an element of the game that is created outside of the games engine and which has been imported into the engine. The asset may be a visible object that makes up part of the game environment, or it may be a sound effect that plays at certain times during the game. Game assets can be 2D or 3D and they can be graphical, audio or **triggers**. The assets are the building blocks of the game, that is, the game objects.

Key terms

Trigger – invisible collisions in a level that will prompt (trigger) a function or event.

Sprites – images used to represent characters and objects in a 2D game.

Gameplay features

A game's design has to cover all elements of the gameplay features. The GDD should cover all of these elements and ensure that the developers understand exactly how everything should work in a game. The following sections outline the different features of gameplay that are designed for a game.

Interaction model

How is the player interacting with the game? Is it through an avatar, that is, a virtual character who represents the player, or is it through some omnipresence, an invisible hand of god, which controls events in the game and of which the game world's inhabitants have no awareness? The type of interaction sets a variety of design decisions, such as the position and viewpoint of the game's camera, the way information is delivered to the player and how NPCs are interacted with.

Case study

Forest Jump

Consider a 2D platformer game called Forest Jump where a character must jump forever upwards through a forest canopy. The image below shows a scene from the game with labels indicating the types of asset involved.

The character is an asset made up of animated **sprites** that make her appear to jump and run.

The background is a static asset showing the forest background which never changes.

The trees are sprite assets and there are branch assets that have colliders so that the player can land on them.

Each time the player presses the jump button, a springy sounding audio asset is played and when the player reaches the top of the canopy of trees they pass through an invisible trigger asset that tells the game's code that the level has been successfully completed.

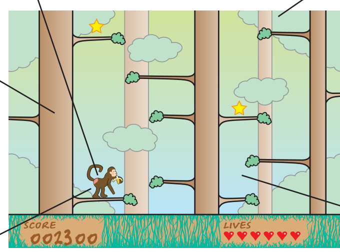


Figure 14.13: A 2D game uses sprites for all its assets

The assets are created using various different tools. 2D assets are created in Photoshop or other graphical imaging software. The GDD needs to generate enough concept art for the graphic designers to be able to create the assets. There is usually a full asset list in the appendix of a GDD, but additional small environmental assets are often added during development as long as they conform to the overall visual style of the game.

Figure 14.14: Examples of 3D assets

Participation

What exactly is the player taking part in? Is it a single player campaign where they follow a story alone? Is it a multiplayer game and, if so, is it a local multiplayer or online multiplayer game? If it is an online multiplayer game, is it a player vs environment (PvE) game where groups of people work together against AI enemies or is it a player vs player (PvP) game where they battle each other in different modes? These decisions will radically change the design of the game. A network programming team is needed for multiplayer games in order to write the code and set up game servers that players will connect to.

Narrative

Does the game have a story? If so, then is it a linear story which will have the same events and outcomes every time, or is it a story with branching narrative where the player's choices affect the outcome of the story? Branching narrative requires multiple scenes and levels to be designed, some of which may never be seen by the player depending on what decisions they make. A story treatment is written as part of the GDD and then, if the game has dialogue, a script will be written for each part of the

game. The designers need to decide if the game dialogue is spoken, in which case voice actors must be hired, or onscreen text needs to be produced. While players prefer audio dialogue to having to read text onscreen, it makes it difficult to release the game in multiple countries around the world because, for every different language region that you want to release your game in, all the dialogue has to be translated and re-recorded. This is part of a process called localisation and can be very expensive.

Game setting

Lots of decisions have to be made about the setting of a game and some of these may affect what genre the game is marketed as. The physical setting refers to how the game world is made up. Is it 2D or 3D? How are objects scaled in the world? What are the boundaries and are there invisible walls to stop the player wandering off the path?

Next the temporal setting needs to be decided. When does this game exist in time? Is it in our past, our present or our future? Once that has been established, we need to know how quickly time is going to pass in the game. Is it real time where every game minute is a real-world minute or will time speed up and slow down depending on the events unfolding?

The environmental setting decides where the game is set. Is it in our world or a fantasy world? Is it on a distant planet or inside a microscopic universe? The environmental setting can also dictate what the world is like in theme or culture. It could be a post-apocalyptic world ravaged by zombies or it could be a world under the sea at risk from an environmental threat. Lots of settings in games are repeated from title to title but elements of originality can still be found even in the most clichéd of settings.

We are used to novels and films telling us stories of emotional highs and lows such as love, loss and betrayal. Games are able to take on an even greater emotional journey by letting us make our own decisions about what happens in the story: this is the emotional setting. What can be more heart breaking than watching your character lose the one they love when it was your decision-making that led them to this loss? Games that manage to immerse us in their worlds and combine this with great emotional plots are always well received, and are responsible for moving games forward as an art form rather than just pure entertainment.

Ethical decisions are also a cause of great emotion in games. The designers may choose to give a game an ethical setting by putting the player in a situation where they have to make an ethical choice. For example, it could be between saving one life or a million, or it could be whether or not to perform criminal activities to further a cause but, whatever it is, ethical decision-making can make games incredibly immersive and rewarding. A good example of this is Papers, Please™, a 2D job simulator where you take on the role of an immigration office official checking who is allowed into the country. The game involves a series of ethical dilemmas where the player must either risk losing their job for a deserving stranger, or feed their own family.

Goals

The designers decide the main purpose or objective of a game and players will not engage with a game unless it has a point. The designers will decide how the goals are broken down into different levels within the game. Often there is an item that the player has to retrieve or a fellow character who needs rescuing, and this is the goal of the game.

Challenges

The challenges that the player must overcome in order to achieve their goals are the next focus of the designers. What are the actual threats or hazards that the player must face? This stage of the design will see the creation and allocation of NPC enemies to different areas in the game as well as the design of environmental hazards, such as falling floors or jets of fire, which need to be circumnavigated to get to the goal.

Rewards

The designers will come up with a list of potential rewards for the player. One of the most commonly found game rewards are experience points or XP. Players will receive XP for defeating enemies or using a particular skill effectively. Once they have enough XP they may be able to spend it on levelling up or new skills, depending on how the game is designed. Other rewards could be unlockable areas within the game, or, perhaps, additional time or a temporary power-up that can be saved and spent in a future level.

Player actions

What can the player do? Can their character perform basic movements like walk, run and jump? Do they have a more sophisticated set of skills such as magic spells, wielding powerful weapons or teleporting across worlds? All these decisions, while easily made, have far-reaching consequences. For instance, if you give the character the ability to fly, what is to stop them from just flying right to the end of the level over the heads of all the enemies that have been put in their path? Player actions have to be designed to give the player the tools they need to complete the game, but not too easily. Some game designers will not introduce all of the player's abilities at the start of the game, but allow them to be gained, one by one, as the rewards of different levels. There is a style of

game design called Metroidvania where the entirety of the game world is available from the very beginning but some parts are only accessible after certain tools or abilities are won by the player. The name comes from the classic games, Metroid and Castlevania, which had this design.

Rules

Once a character's abilities are decided, the rules of the game must follow. If the player is able to make their character jump, then how high? Can they double jump? Wall jump? The movements of a player significantly affect their progression through the game world, and the movement rules need to be carefully considered. A player's valid moves are the ones that they are allowed to make and the game should ensure that no invalid moves are made accidentally. The rules of a game world do not just apply to the player character, they also apply to enemies, and they decide the physics of the game world as well. If a character has magic, for instance, is it limitless or will they have to top it up with something like mana potions? If that is the case, then how many mana potions should there be per level, and how many can a character hold? All this has to be designed and decided before building the game.

Feedback

The way in which a player is aware of their progress is crucial in a game. The way in which software communicates with humans is a field of study called **human computer interaction (HCI)** and it is never more crucial than in games design. The last thing you want as a player is for your attention to be taken off your object of focus in a game. If you are fighting a boss, you do not want an indicator to start flashing in the middle of the screen and block your view. Games use a head-up display (HUD) to show information in a discrete and unobtrusive way. The more subtly information is fed back to players, the better the feedback, although it does need to be clear.

> **Key term**
>
> **Human computer interaction (HCI)** – the study of how people interact with machines, and the best possible ways to design interfaces between people and machines.

Difficulty

The difficulty of a game, or degree of challenge, sets out how hard it is going to be for a player to complete the game. Many games use a difficulty curve where the game starts easily and gets progressively more difficult the further through it the player gets. This may be actioned through enemy strength or a number of hazards in a level. Many games allow the player to set the overall level of difficulty at the start of the game depending on their confidence, and some games will release a new difficulty level once the game has been completed on its hardest setting.

Game mechanics

The mechanics of the game include all of the functional game elements that need to be designed and are taught to the player, through either a tutorial level, dialogue or trial and error. Game mechanics may include an inventory system that lets players collect items, a crafting system that combines items to create new ones, or a scoring system whereby different accomplishments generate higher scores and win conditions. A win condition is the rule that sets the circumstances in which the player wins either an individual level or an entire game. In the classic arcade game Donkey Kong™, the win condition is reaching the princess at the top of the scaffold tower. Other titles, such as strategy games, may have more complex win conditions, whereby you must beat the enemy within a certain time frame using only particular resources.

Game structure

The actual structure of the game includes the number of levels, **cut scenes**, enemies and the progression that the player must make to complete the game. This is shown in a number of design documents, such as storyboards which show simple drawings of the sequence of events in a game, flowcharts, diagrams that explain how different rules or algorithms will work, and activity diagrams that show the way in which a player navigates the game from the opening menu to the final credits.

> **Key term**
>
> **Cut scene** – a cinematic sequence in a game that tells part of the story. Cut scenes can be a separate movie clip or can be shown during a level.

Quality

Games should be designed in such a way as to ensure that they are as high quality as possible, otherwise they will receive poor scores when reviewed and sales will be low. A game's compatibility with its platform needs to be considered during the design phase. For instance, if a game is being made for a touch-screen device, how many different buttons will the player be able to cope with at once? If a game is being developed for multiple platforms, then it must not rely on specific platform features that are not common to all of the platforms it is being designed for.

The performance of a game is also a consideration, as the designers making a game for a device with a lower specification GPU and CPU would not be able to design photorealistic graphics and hours of cut scenes with audio dialogue, because the system would not be able to handle it.

In addition, the designers have to consider the gaming experience at all times during the design stage. Why would people play this game? What is unique about it? How does the level being designed compare to the previous level? Does it have anything new or interesting for the player to engage with? Bad games are games where these kinds of questions have not been asked or answered, and where the player is not considered enough.

Design documentation

The game design documentation (GDD) is passed from team to team throughout the games development process. It is of crucial importance and will be referred to constantly by artists, asset creators, programmers and testers.

Audience, purpose and client requirements

The GDD must begin with an overview of the requirements that the client has set out. The client will be the publisher for most games, but for crowdfunded games it will be the backers. Their requirements may include a series of milestones, at certain dates, that the client wants you to meet, or a particular emotion or atmosphere that they want the game to evoke.

The design will then go on to specify the audience requirements, that is, who exactly is the game for and why would they want to play it? Pinning down the target audience is crucial; it is nearly impossible to design a single game that will cater for everyone. There are so many different types of player buying games today and the game designers have to be acutely aware of this. Almost every decision they make when preparing the design document should consider the audience and, because of this factor, it must open with a very clear specification of whom the game is for.

The audience for a game is worked out by looking at various demographics, that is, categories of people based on their age, gender, spending power, education, location and family status. The amount of time that they spend playing games and their gaming experience is also important. You could design a simple platform game in the style of Super Mario Brothers™ for a child based on simple game mechanics, but you could also design a similar game for a 30–40-year-old who had enjoyed the original games and wanted to enjoy a nostalgic experience. A game studio might identify its audience based on similar titles that a particular audience has previously enjoyed, sometimes in a bid to steal market share from a competing publisher or platform.

There may also be a specific purpose to a game. While most games are designed purely for entertainment purposes, sometimes a developer will create a game that is designed to introduce a new peripheral device, such as a motion controller, educate the player in how to train their brain or learn to play an instrument, or help to advertise a new film or TV series. Creating games for marketing purposes, nearly always creates weak games because their design and development is rushed so that they can be released in time to advertise the new film or TV series that they are tied into. If you wait too long to release a game for marketing purposes, then all interest in that film/TV series diminishes, so games are rushed out to market without being properly designed or tested. It is only when the game is begun at the early stages of a film or TV series' development that it stands a chance of being a quality title.

Case study

A sample client brief

Frodo Games, an international publisher, has asked your development studio to make a game that serves as a prequel to the popular children's TV series *Seb's Teds* which follows the adventures of a young boy and his teddy bears. The TV production company that owns the rights to *Seb's Teds* are letting you use the names and images from the show on the understanding that the game will contain no violent scenes, does not portray Seb or any of the Teds acting aggressively, and maintains the ethics and morals that the show is known for.

- What game mechanics would you consider appropriate for this title?
- Write a description of the audience for this game.

Legal and ethical considerations

There are several laws covering the design and production of computer games and these need to be considered when looking at a game's design.

Copyright

The most important law to consider is the copyright law that protects creative works made by an individual or a company and stops other people from copying them and making money from the copies. The copyright laws cover game content such as characters, places and specific game designs, but the creation of clones in gaming is as old as the games industry itself. Consider the mobile hit Flappy Bird™. As soon as that became popular, it was very quickly followed by Flappy Shark™, Flappy Duck ™ etc, which are clones of the original game with just enough difference that the developers could not be sued for breach of copyright. The trouble is that the law does not protect the look and feel of a game; only the intellectual creation itself is protected so many popular games are cloned and rebranded. However, the developers doing the copying will not have access to the original game code so it will always be an approximation of the original instead of a complete copy. Part of the problem is that it is argued that the functionality of one game is very much like another or similar to hundreds of other titles that have a small element of the game in question. If so many characters run, jump and shoot, would the creator of the first game that contained a character running claim that all other games have stolen their copyright? Games contain lots of elements that fall under copyright protection, for example the images, titles, names and the design of the levels themselves. All this comes under the term intellectual property (IP), which is used to describe the content within a game as well as the game itself. You might hear a publisher bragging about a 'brand new IP' which means a new title that has not had any previous games out and is all-original. A game tied to a film cannot claim to be a new IP, because the film has the IP; neither can a sequel.

Consumer rights

As computer games become a common part of our daily lives, existing laws have been adapted to consider games. One of these laws is the Consumer Rights Act 2015, which gives customers stronger rights as the consumers of (ie the people who buy) video games. UK residents can now claim a refund or repair if the digital content that they have bought, that is the game, is not working or is of unsatisfactory quality. Poor quality is often a consequence of a publisher rushing the development to meet deadlines and not testing the game properly. Interestingly, this does not just cover digital games that need to be paid for, but it also covers disc-based games and games that are free to download. Therefore freemium titles also have to be of a decent quality or the developers are obliged to fix the defective product. This is a good example of the law changing to include new products and patterns in society. Game designers have to stay up to speed with the constant changes in law and the differences between the laws of different countries.

Licence fees

Games designers have to consider other laws when releasing their games and a lot of these are to do with the financial side of games development. If a game is going to be released on a platform such as a console, they will have to pay a licence fee to the console manufacturer to get their game released for that system. Therefore, if a game is made for PlayStation4® 4, then the developer would have to pay Sony® a licence fee. This would be paid on the understanding that the game was of sufficient quality to appear on Sony®'s system as the company's reputation would be damaged if inferior products were released. This agreement would be made using a legal contract that sets out the responsibilities and actions of both parties.

Royalties

Another financial consideration is the payment of royalties. If a game is made using a commercial engine such as

PAUSE POINT

Have a look at the blog www.gamerlaw.co.uk

Now imagine that you have created an amazing new game that is unique in its story, features and visual style.

Find out how you would be protected by copyright law.

Hint Copyright law protects 'the expression of an idea' but not 'the idea itself'. How will this protect you?

Extend What will happen if you release your game in different countries?

Unreal Engine, there may be an understanding that games can be published for free, but, if they earn over a certain amount in profit, then they must start giving a percentage of the game's profits to the commercial engine. Music within a game will also be subject to royalty payments in the same way that it is for a film or TV show. In terms of royalties for the actual developers, they tend to be paid an agreed fee at certain stages of the game's development and, after it has been released, they do not have any claim to the profits. A few games publishers do offer royalties to developers, but it is not commonplace.

Digital rights management

Digital rights management (DRM) is another legal consideration for games developers. This is the area of law that concerns itself with how digital products are used and shared. Once a game has been purchased as a download, is it acceptable to copy this to other devices in the same household or to give copies to friends, or even sell copies? If you bought a DVD of a game you could lend it to friends and later sell it on, but you would only ever have the one copy. Different companies take different approaches to DRM. Apple® will let you set up family networks that share apps, while Google™ lets you log into as many different devices with your Android™ account as you want. Steam™ forces you to authenticate whichever machine you log into with Steam™ and, if another user is on there, you can request access to their game library. Microsoft® originally announced that the Xbox® One® would have a constant internet connection to check the DRM of any game played. This was introduced so that they could monitor which games people were playing and for how long. They tried to spin it to customers as a way to get constant updates, but this received such a backlash from players that they quickly changed their mind. Their competitor Sony® introduced a 'virtual couch' that lets friends play each other's games whenever they are both online.

Ethical considerations

No specific law covers the ethical considerations of games design, but it is something that games designers have to think long and hard about. If they are representing a particular country or culture, are they doing it in a way that is ethically acceptable? (For example, they should not suggest that all the people from a particular religion are violent. They cannot discriminate or portray stereotypes in games.) If they are putting the player in a questionable situation, are they clear that they appreciate that this is an ethical question and not something that the player should just accept? Questions about violence and sex in video games are always present in the media and many games have been accused of provoking all sorts of real-world crimes. Therefore games designers have to think very carefully about what they are depicting in their games and who could be influenced by them.

Game design

The following features should be included in the design documentation for a game.

- Type of gameplay – a definition of what the game involves, e.g. FPS, MMO, RPG.
- Data dictionary – a table showing all the data used in the game, its type (e.g. number or text) and its purpose.
- Algorithm design – designs for the game's functions and features using **pseudocode**.

Key term

Pseudocode – a way of writing code without it being in a specific programming language.

- Storyboards, flowcharts and an activity diagram – all show the structure of a game, how the game should look and flow.
- Visual style:
  - world – the type of terrain, style of architecture and style of objects
  - characters – player, enemies and other NPCs
  - feedback interface – how information is communicated to the player
  - perspective – 2D, 3D, first-person, third-person, scrolling, aerial, context-sensitive.
- Full motion video – used for cut scenes and menus.
- Asset lists – graphical, audio, video.
- Gameplay features – what features make this game original and unique.

Link

For more on gameplay features, see the section Gameplay features.

Worked example: Hazman

Here is a sample of the game design for a strategy game called Hazman. (The full design documentation would be a lot longer and more detailed.)

Type of gameplay – a point and click strategy game for PC that sees a young hero clearing chemical spills in industrial areas around the world. As the game goes on, Hazman starts to suspect that these incidents are not accidents after all and a conspiracy theory soon leads to a dramatic rescue.

The player will quickly assess an area and race against the clock to find the right solutions to clearing up the different spillages. The gameplay is a combination of solving coded puzzles by mixing the right chemicals together and 'escape-the-room' style puzzles to get the hero and any good NPCs to safety.

Data dictionary – each level will have the following data requirements.

Name	Data type	Data length	Scope	Purpose
LevelNumber	Integer	1–99	Public	Stores the current level number that the player has reached
MovementSpeedMax	Float	0–25	Public	Stores the current movement speed based on input from the player
PlayerScore	Integer	0–4	Public	Stores the current score of the player
ChemicalTypes	Integer	0–9	Public	Stores the amount of different chemicals that can be spilled in the current level
ExitPosition	Vector3	(0–999,0–999,0–50)	Public	Stores the position of the level exit

Algorithm design – the game level will be split into tiles, with 4 rows by 4 columns. Each spilled chemical will start on one tile and, as the game progresses, will spread onto more tiles until the player becomes stranded.

Pseudocode:

```
LevelGridArray{0000,0000,0000,0000};
On Game Start
  While ChemicalTypes>0 – Set random grid element to 1, ChemicalTypes--
On Game Loop
  For Each Chemical – Spread to next available empty grid space till array full
```

Storyboards – the opening cut scene for the game will be a series of still cartoon panes that tell the story of the first chemical spills. Additional cut scenes will be made after release, available as downloadable content.

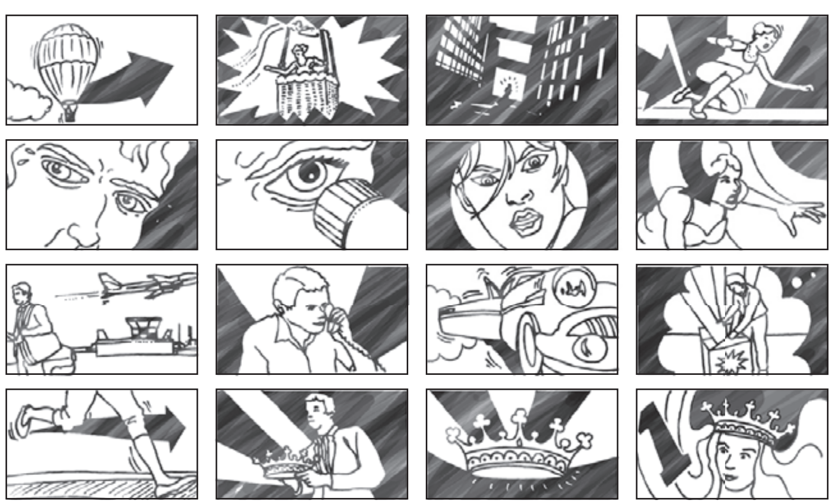


Figure 14.15: A storyboard is a visual plan of how aspects of the game should look, usually showing cut scenes

Visual style – the game's style will be highly stylised cartoon visuals with thick black outlines around all assets. All floor tiles will have simplified textures and the chemicals will have a unique colour that is bright and has a shiny surface. The characters will only be seen from above and they will have exaggerated proportions (large heads and eyes, a small torso, short limbs and large feet). The HUD will display the time prominently on the top right and messages will be presented to the player in the centre of the screen (gameplay will pause until the message is discarded by the player).

The game will have a top down perspective throughout gameplay where the player clicks on different tiles to activate the puzzles and solutions. Where a solution is a success, a **context-sensitive perspective** will zoom in on the player's character whose animation will change to celebration mode.

Full motion video – there will be a cut scene at the start and end of the game. This will be an MP4 containing static images and the game's theme tune.

Asset Lists

Graphical	Audio	Video
Player sprites	Level specific theme tune	Opening cut scene
Floor tile sprites	Spillage Sound Effect (SFX)	Closing cut scene
Wall sprites	Creeping chemical SFX	
Hazard sprites	Death SFX	
Crates sprite	Win SFX	
NPC sprites	Time running low SFX	
Chemical sprites	Countdown SFX	
Menu textures		
HUD textures		

Gameplay features – the game is puzzle based. The first four levels will contain chemical puzzles where each chemical has an antidote, made up of ingredients that Hazman must collect, which will be spread around the level. Once these ingredients are mixed together, the level will be complete. The next four levels will involve escape-the-room puzzles where there are no antidote ingredients, but Hazman must assemble items to escape from the room. For instance, he will need to get a key from a monkey in a cage, and, once he opens the cage with a crowbar, the monkey will jump to a higher shelf out of reach, Hazman must then find a banana to tempt the monkey down and retrieve the key.

Key term

Context-sensitive perspective – this is when the in-game camera changes to focus on something that the player has selected or a change that they have caused.

Development choices

There are many different decisions that the development team need to make as they embark upon the creation of a computer game.

Platforms and programming languages

A number of different factors decide how a game is made and one of the most crucial of these is which platform it is being developed for. If the developer knows in advance that the game is only going to be released on one specific platform, then they will be able to find out the language that the platform is designed to be used with along with all the different games engines that can be used to create games for that platform. If they know that they are going to release the game onto multiple platforms, then they must use a games engine that supports all of these different platforms, or risk increasing the development time, because they would have to start from the beginning for each intended platform.

If you are writing games purely in code, then an integrated development environment (IDE) is used. This is a system, such as Visual Studio®, which supports one or more programming languages and is able to find errors in the program and, once finished, package the code into an executable application.

A games engine will use an IDE to write the code portions of the game or it will have the IDE built in. For example, a console game that is going to be exclusively for the PlayStation® 4 could be written in C++ and all of the input and output will be designed specifically for the PlayStation® 4. It could be written in the Visual Studio® IDE. It is not possible to build levels and arrange assets visually in an IDE alone, which is why games engines are used.

If a game were going to be released on Windows® operating system PC, Mac® and Linux, then it would be best to create it on a games engine that supported all three platforms as it would be difficult to write the game code to work on three different operating systems directly in code.

Some programmers are able to use a simple text editor to write code and make games. In the 1980s, it was possible to buy magazines which contained code samples for players to copy to create their own games on systems such as the ZX Spectrum or Commodore 64.

C++ is the most widely used programming language for games. C# is a derivative of C++ but it is missing some features and does not run as fast. However, it is very useful for writing game mechanics for engines such as Unity® software because it is easier to understand, but still powerful. JavaScript is another object-oriented language, which can run slowly due to the way it is translated for any platform. It has not been embraced by games console developers for this reason, but it is the best way to write programs for Android™ without using a games engine, because the Android™ system is based on JavaScript. Scripting languages like JavaScript and HTML5 are widely used for making web games because they are quick to run and small in file size. Older languages such as Python® or Delphi® are not really used in professional games development but are very useful for learning how to program.

Application programming interfaces

The design of the game should also include reference to any application programming interfaces (APIs) that are going to be used. An API is a code library that has been written for use in other programs in order to quickly add extra functionality. It may be that the game needs to communicate with an external service, such as Apple®'s iOS Game Center for Developers to take advantage of online leaderboards, or it could be that the developers want to add a virtual reality option so the Oculus Rift™ API might be used. When a programmer wants to use an API, they reference it at the top of the code where it is needed in an 'Include or Using statement'. After the API has been included, the programmer can reference it whenever they want during development.

Computer game development kits

If a game is being designed for PC or web, then it is quite straightforward to test the code during development and the designers will ensure that the correct hardware is present before the build begins. If the game is being built for a console, however, the development team will need to buy a console development kit for whichever platform they are designing it for. A console development kit is a version of the console that can connect to a PC and have test builds of the game deployed directly onto the console, to test how it works on the system for which it is being designed. This is the best way to test a game for a console and it ensures that all of the controls work, that the game loads and saves correctly and that any communication with the console's online capabilities work correctly. Some console manufacturers sell development kits while others are willing to lend them out to developers, especially ones they have worked with before.

Intended platform/media for delivery

An important question for developers to ask themselves considers which features of the intended platform the game will take advantage of. You may want to keep all the game's features common to all platforms or have specific functionality designed for individual platforms. These features should be included throughout the game, and should also be promoted in the marketing for the game as unique selling points.

In addition, the intended rating of the game needs to be considered during the design process (see Table 14.3). If a game is going to have a PEGI rating of 7 then it may have violence but the violence must be unrealistic (ie something that could not happen in real life, like a magic spell or a piano falling on someone and them recovering instantly). The rating makes a huge difference to a game's design and everyone involved in the design needs to know what the intended rating is so that they can make their designs age appropriate.

Another consideration is whether a game will be delivered as a digital download or a boxed game. If it will be released as a digital download, then there needs to be a series of graphical assets created for the game's entry on the download page. Digital platforms will also ask for a number of differently sized icons: these are graphics that can be resized, and graphics that can be used as part of a marketing promotion or sale (boxed games also need graphics for online advertising). All of these images have to be provided by the game's development team. If it is going to be a boxed game, then it also needs to have box art drawn up, an image to go on the printed disk and any instructions or promotional codes that are going to be included inside the game box.

Table 14.3: PEGI games ratings, for more information visit **www.pegi.info**

3	Suitable for ages 3 and older. May contain very mild violence in an appropriate context for younger children, but neither bad language nor frightening content is allowed.
7	Suitable for ages 7 and older. May contain mild or unrealistic violence (eg violence in a cartoon context), or elements that can be frightening to younger children.
12	Suitable for ages 12 and older. May contain violence in either a fantasy context or a sporting action, profanity, mild sexual references or innuendo, or gambling.
16	Suitable for ages 16 and older. May contain explicit or realistic-looking violence, strong language, sexual references or content, gambling, or encouragement of drug use.
18	Unsuitable for persons under 18. May contain extreme or graphic violence, including 'violence towards defenceless people' and 'multiple, motiveless killing', strong language, strong sexual content, gambling, drug glamorisation, or discrimination.

Animation timeline

Different parts of a game will involve various different animated sequences. Animated characters will have a series of animation cycles that they will play on repeat when the character performs different actions. For example, walk cycles, run cycles, idle cycles and so on. The artists will draw single frame images of each position needed in the animation.

Cut scenes within a game involve assets used within the game, so the artists will create storyboards that follow the cut scene and may create a timeline showing the order in which assets are needed for that scene.

Production schedule

The schedule for a game is recorded in a number of different ways. The project milestones are set up first. The milestones are a list of stages that the project will go through, and the dates by which they need to happen. The art stage is followed by asset creation, mechanics implementation, level building and testing. Completion of these milestones may trigger the publisher to pay the development studio a portion of their fee, while any delays could result in a reduction in payment.

The GDD will contain a more detailed breakdown of the schedule. This may be a Gantt chart, which is a kind of bar chart that displays all the different tasks that need to be carried out to complete the project, who is doing them and when they need to be completed. The team may use project management software to create and share the schedule. It is usually a live document that is regularly updated and adapted based on changes during development.

Resources

Games cannot be designed without various pieces of hardware and software and this can be one of the biggest expenses for a games studio, especially if they are making a game for the first time. Every developer in the team will need access to a PC and the appropriate software. 3D modelling software is very expensive, as is graphic design software, but they can be purchased on a monthly basis which means that the studio only has to pay for what they need. However, this is not a good long-term approach. The PCs required for games development need to have a good enough hardware specification to be able to model and animate in 3D.

Graphic designers and concept artists will need additional resources such as graphics tablets in order to complete their tasks (see Figure 14.16). They will also require traditional drawing equipment (pencils, pens and paper) and scanners to make these images digital so that they can be shared.

Figure 14.16: A graphics tablet is an invaluable tool which provides a lot more accuracy than a mouse

In addition to these resources, a games studio will need a building to work in, and they will need to pay for utilities such as electricity, gas and broadband internet to function as a business.

Test plans

After a game has been created, it must be tested thoroughly. Players will not be happy if they have paid for something that does not work properly. Testing is taken very seriously and, while a lot of people imagine it to be tremendous fun to play games for a living, professional games testers need to be able to understand both the design documentation and the programming code because this is what they will use as a reference when

they are creating and executing their tests. The test plans they write need to check the playability, performance and quality characteristics of the game.

There are a number of different strategies that can be used when planning the testing of a game, and the producer or project manager will be responsible for employing a test manager who will then choose the most appropriate tests. There are various types of testing documentation that can be used to plan the tests and then record the results.

The test plan covers the development of the entire game and makes sure that there are no flaws in the product that is eventually released. This is a very large-scale task and if any problems are found during testing the test plan will ensure that they are recorded and fixed. Games that are released with flaws or errors in them do not sell very well and receive poor reviews from games journalists.

Each test will have a test case, which is a particular scenario that the tester will undertake to try out a feature of the game, or a command, under certain conditions. These conditions might be the direction of movement or the number of items in an inventory. It is important that the test cases cover as many different scenarios as possible so that there is nothing that the player could do when the game is sold that has not already been tried during a test.

The test plan is used as a test log, where all of the outcomes are recorded and an indication of what needs to be done to correct any errors is added. If a problem is found, the tester may take a screenshot to use as test evidence. This screenshot is then passed on to the developer who will use it to get a clear idea of the problem and then fix it. Once all of the testing is complete, a test report is written to show where problems were found, what has been fixed, and to summarise any vulnerabilities in the software or issues that might resurface. Once a game has been tested to a satisfactory level, it will then be released for sale.

Case study

Working out mesh constraints

The official documentation for Unreal Engine 4 states that the maximum vertex count for a level in a mobile game is 65,000 vertices per mesh. So what does this mean? A 3D object in a game is created in software such as Autodesk® 3D Studio Max and is referred to as a mesh. Meshes are made up of single points called vertices and every vertex in a virtual world has to be calculated and positioned. If a simple cube mesh has 8 corners, then it has 8 vertices. If you divide the maximum possible amount of vertices, 65,000, by 8, you get 8125. This means that you could only have 8125 cubes in a mobile game.

It does not seem like that big a deal until you consider that a typical console character can be over 80,000 vertices: this is too many for a mobile game. Any character in a game will instantly increase the vertex count, especially if they are animated, as this requires more vertices in order to stop the mesh looking strange when it moves.

The next consideration is the number of bones used in animated characters. A standard human skeleton rig in an animated 3D character has 60 bones. The upper limit for bones in a mobile game according to the Unreal® Engine 4 documentation is 75 per mesh. This means that if you use the typical skeleton rig for animated characters, then you may not have enough bones left to have any other animated characters in the game.

Therefore designers working on mobile games will do a number of things to make sure that they can have multiple animated characters in their games. First, they will design the game with a very basic art style so that the meshes use as few vertices as possible. The more curves an object has, the more vertices it will require. So designers will create objects and characters with as many straight edges as possible. If you look at 3D games on old consoles, you will notice how the designers have avoided creating curves because they were limited in the number of vertices they could have.

Developers will also use level of detail (LOD) groups. This is a process where a simpler version of a 3D model is put into a game level when it is far away from the player, but, when you get closer to it, a more detailed version of the mesh is swapped in. For example, the less detailed version of a house may be a flat cube with images of windows and doors, but, when the more detailed version is swapped in, the house has doorframes, handles, window ledges and window frames.

Finally, developers can design their characters so that they are either not animated at all (they could still rotate or move), or they have simple skeletal rigs (no finger bones or feet bones, for example). Characters in mobile games tend to have cube-shaped feet and hands, which developers tend to get away with because these characters are viewed on small mobile device screens.

Constraints

The constraints of a platform are the limitations that designers have to consider when producing the design documentation. These limitations can have far-reaching effects on how the game is going to play.

Platform limitations – hardware

Perhaps the biggest concern for designers is what the hardware can cope with. A high specification gaming PC with an Intel® i7 processor, 16GB of RAM and a powerful graphics card will not have any difficulty running any game that is thrown at it but if a game is being designed for a less powerful platform then the abilities of the hardware will directly affect the game design.

Due to copyright we are unable to reproduce this content here

Figure 14.17: A 3D character designed for animation. Screenshot provided by Dan Bennett

Platform limitations – software

The software on a system does not produce as many constraints to games designing as the hardware does, but the version of an operating system that is being used may cause some problems. Mobile operating systems are updated all the time and if a game is not compatible with a few older versions then the designers will lose a portion of their target audience. Apple® fans often complain that new system updates make their hardware redundant and force them to upgrade, and that this also reduces the number of games available to them because a newer game, marked on the app store for a higher iOS version, will not even appear to them.

Reviewing and refining designs

Producing the design documentation for a game is not something that you do once per game and then make no changes. A game's design should be reviewed as often as possible, and the more time set aside in the schedule for looking at the designs and for seeing if they can be improved, the better.

Primary feedback

After the initial design documentation is completed, a game's designs will be shown to clients and other interested parties, potential players or any licence partners, to see what they think of them. They will be asked about the quality, effectiveness and appropriateness of the designs for the intended audience. Sometimes, a number of **prototypes** are made as part of the design process and these will be demonstrated to potential users via the internet or at gaming conventions. Designers have to be careful about showing too much too soon because a lot can change during the design stage. However, much can be gained from knowing what your potential audience thinks about your ideas. It provides you with feedback with which to review and refine your designs.

Key term

Prototypes – small test game levels used to make sure that the key features of the game are working and to illustrate to clients and potential customers what a game will be like.

Client communication

Prompt and professional communication with the client is crucial for the design team to show that they are on top of the project and that they are not hiding any slipped deadlines or major problems. Whether by email for brief exchanges, or face-to-face with the client for important discussions, the design team has to be in regular communication with their client. Many games studios subcontract some areas of the design and development to different studios and this can cause a breakdown in communication which may result in a weak final product. In 2013, a highly anticipated game received very low review scores and disappointed the gaming public because it did not contain scenes that had been in the promotional material. A group of gamers took the developers and publishers to court. The publisher blamed the developer but the developer claimed that it was the responsibility of the publisher. It is possible that these problems could have been avoided if there had been better communication between the developers and the publisher.

Meetings and timescales

Whenever different teams within a developer are working together, and when they are working with a publisher or subcontracting elements of the development, it is important to have regular meetings and to keep records of these meetings in the form of minutes, decision logs and action lists. In the games industry, one form of **project management** used is called Agile Scrum. This approach

requires the teams working together on a creative project to meet every day for 'daily scrums' where they discuss the current progress and identify any changes that need making.

Key term

Project management – different methods and procedures used to keep a project on track and under budget by minimising and mitigating risks and issues.

When changes to the designs are made, either due to feedback or because of review work within the design team, the timescales for the overall development must be updated to include how long the new changes will take to implement.

Updating the design documentation based on feedback and reviews means that a new copy must be distributed to the entire development team as soon as the changes are approved. Alternatively, a live document can be edited which everyone has access to.

Refining ideas and solutions

When the design for a game is started everything is possible, but then, necessarily, it is slowly refined down into a more specific and sophisticated design that players will want to play. Through making prototypes, it is possible to see which ideas work or are popular, and then to refine them down to the best possible version. The game's designers will choose their refinements by selecting the most successful (popular or effective) features, and by removing elements that proved too difficult to get working properly in the prototypes. The game's publishers and potential customers will be shown the prototypes and asked for feedback, which will then inform their ideas. Problems in the game's design that are revealed by prototypes require solutions which, once found, are then included in the design refinements. It is hard to find out which features were cut out of popular games as the games studios tend to keep these details to themselves, but the internet is full of rumours of features that might have been.

Theory into practice

Media and communication skills

The games industry is at the forefront of creative media and technology. Given this, its working practices tend to be driven forward by dynamic individuals and cutting edge software. You have to be able to use excellent communication skills to ensure that your role in the game's design and development is as efficient and effective as possible. Below are lists of the written and verbal skills required to work successfully in the games industry.

Written skills:

- be able to use email to share or request information
- create design documents that colleagues can easily interpret
- write reports detailing a project's progress for clients
- create presentations with visual aids to assist understanding.

Verbal skills:

- be able to communicate effectively one-to-one with colleagues and subordinates
- be able to communicate and work effectively as part of a team
- be able to communicate effectively in both informal and formal situations.

How effective your verbal skills are will depend on how well you use tone and body language, as well as what you actually say, to remain professional and convey information. This is especially important when you are giving presentations or talking to clients. You have to use positive language and be able to reassure your audience that you are in control and you understand the scale and scope of your presentation topic. You must use the appropriate technical language when talking to people, especially when presenting information across different teams. If you use too much technical jargon with non-technical colleagues, the discussion may be lost on them.

1. You must always consider how you are responding to people. Are you being supportive? Are you making sure that everyone is getting the chance to talk, not just the loudest person at the table? If you are good at resolving arguments or conflict in teams, or managing the expectations of clients, then a leadership role in games development could be in your future. Make a list of ways in which you can develop your written communication skills.
2. Write a list of tips for yourself on how to communicate effectively in the following situations:
   - when discussing the design of a game with teammates
   - when delegating design tasks to subordinates (people that you manage)
   - when presenting the design of a game to clients.

Write down ideas for how you would be supportive to teammates and subordinates in a design team and how you would make sure that everyone's opinions and ideas were heard.

C Develop a computer game to meet client requirements

Programming a computer game can be very different to creating an office software application or a web application but there are still similarities.

Just as a software application uses code to create menu structures and events, so does a game: it can dictate the flow of the game from the loading screen, and it accesses files in similar ways for game saves. When games programmers are deciding how to proceed from the design documents that they have been given, they usually have the first decision made for them: which programming language they should use. This decision tends to be taken out of their hands because the choice of platform will decide how the game is going to be built. There are still some choices available to the programmer but they are narrowed down a lot. Some computer games development companies will have a particular house language that they insist that everyone uses. The implementation of the programming will depend on the language that is being used but also on the complexity of the game.

Principles of computer games development

Before development begins, there is a basic understanding shared amongst the development team about the principles of games development. When working on a game all team members, regardless of their individual role, have to understand every step of the process in order to ensure that they are working at the optimum level.

The design documentation should have been distributed to the entire team before the development phase. Certain technical diagrams, also known as **schematics**, are delivered to the people who are responsible for the production of those elements. If a 3D game has a third-person character, then they will require a 3D mesh that has been rigged, a series of animations, a state machine to control the transitions between animations, space and a player control program that reads input from the player and changes the animations, movement and rotation of the player character accordingly.

Computational processes

Various computational processes are used as part of games development. These processes happen discretely in the background of the development process and during runtime when the game is playing. One example of this is the **rendering engine**, a system that is responsible for changing the massive data set that represents the objects, textures and players in a game and converts them into a series of 2D images that are rapidly changed. We believe we are viewing a world in 2D or 3D and seeing it move but, as with TV, film and traditional animations, we are watching a series of flat images being quickly animated before our eyes. That is why, when a system slows down or **lags**, the images can freeze (we are stuck on one image waiting for the next one to render).

The screenshot below (Figure 14.18) shows how a rendering engine can also add effects on top of the world that we are being shown.

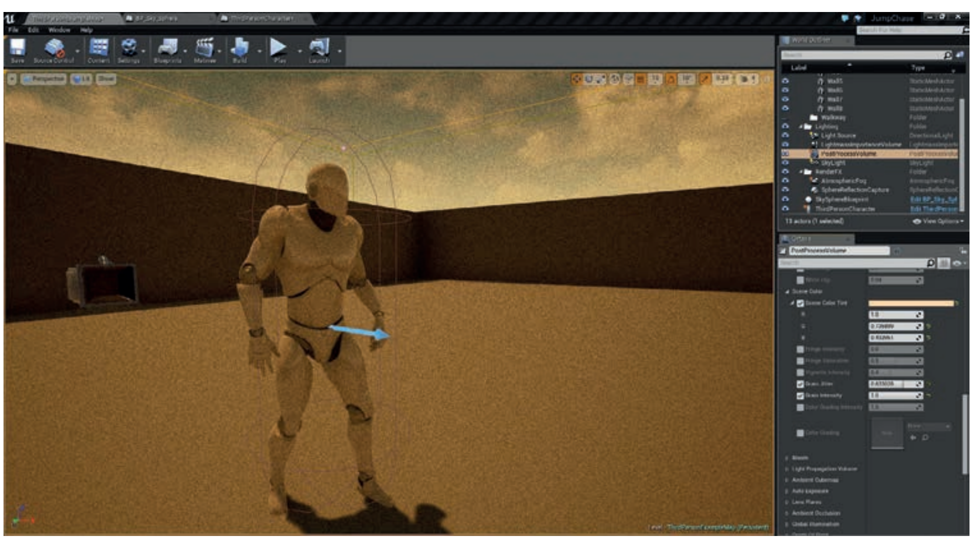


Figure 14.18: Camera effects can be added to a game

Step by step: Production of a third-person 3D game using a games engine

4 Steps

1 The **3D modeller** is given concept art and detailed designs for the player character. As the character seen most in a third-person perspective game, it will have lots of vertices allocated to it. The modeller will understand, based on the animation schematics, how the character will need to move so they will create the mesh with movement in mind. This means that when the animator takes over, they will receive a model that has the rigged bones ready for animation and is modelled in such a way that allows for all the required animation. Imagine that a human character had been modelled and nobody had told the modeller that one of the animations sees the character reveal a tail that they had been hiding. If the tail was not modelled and rigged, it would not be able to feature in the game.

2 The **animator** will use the animation schematics to create all of the different **animation cycles** that have been requested. In **photorealistic games** with big budgets, the animators will use **motion capture** studios, where actors perform the animation movements while wearing motion capture suits. This speeds up development but costs a lot of money.

3 The animator will then hand the animated rigs over to a **technical artist** who will put the animated character into the games engine and create a state machine, which is a system that decides when one animation should change into another, for example when a character should go from standing to walking and from walking to running. The technical artist does not need the animator to explain all of the animations created to them because they can see them in the design documentation and perform their part of the job easily.

4 Once the animations have been completed by the technical artist, they are handed over to the programmer who has been writing code to make the animations change on the push of a button or pressure on an analogue stick. This example is for the production of a third-person 3D game using a games engine. The process would differ depending on the type of game and size of team.

Key terms

Schematics – a technical diagram showing the content and function of game elements.

Rendering engine – the software in a games engine that converts virtual worlds into 2D images which are then animated.

Lag – a delay or reduction in the game's frame rate.

3D Modeller – a job role that involves the creation and texturing of 3D objects.

Animator – a job role that involves the creation of movement in game objects.

Animation cycles – different motions that characters will use on repeat such as running and jumping.

Photorealistic games – games that try to look as lifelike as possible.

Motion capture – a process where real-life motion is converted into data that can allow game characters to move in the same way.

Technical artist – someone who works between the technical and design teams and understands both.

It might be a temporary effect tied into the game's story or an artistic decision that is made later, but post-processing effects can be added to change the overall look and feel of a game. The example in the screenshot shows a film-grain effect being added to make a game seem more 'old world'.

Physics engines are an important computational process that can affect the speed at which a game runs. If a game uses too many physics objects, then the game can slow quite significantly.

Lighting is another crucial computational process and the real-time creation of shadows is a process that can slow a game down significantly or require that it is only played on a very powerful machine. Some PC games allow you to change the quality settings in order to run a game on a slower specification. One of the first things to get turned off will be real-time shadows. As with game physics, something that happens easily in the natural world takes a massive amount of calculations and processes to simulate in a game world. Shadow quality can, usually, be changed. A hard-edged shadow looks unrealistic in most situations, but more realistic than an object that does not cast any

shadows. Soft-edged shadows that fade into the light are a lot more realistic, but require the shadow to be rendered a number of times and therefore take a lot more computational power.

Research

The next time you play a game, pay attention to the shadows and physics in the game world and see if you can notice where developers have taken short cuts. For example, see if the main character has a shadow. Can you see any shadows that are not affected by the main source of light in a game?

Applying mathematics

When a game is being developed, all the objects which require physics will have it applied. This is usually done in a games engine but, when the game is written just using code and assets, the physics will be an API which is then applied to objects that require physics in the code.

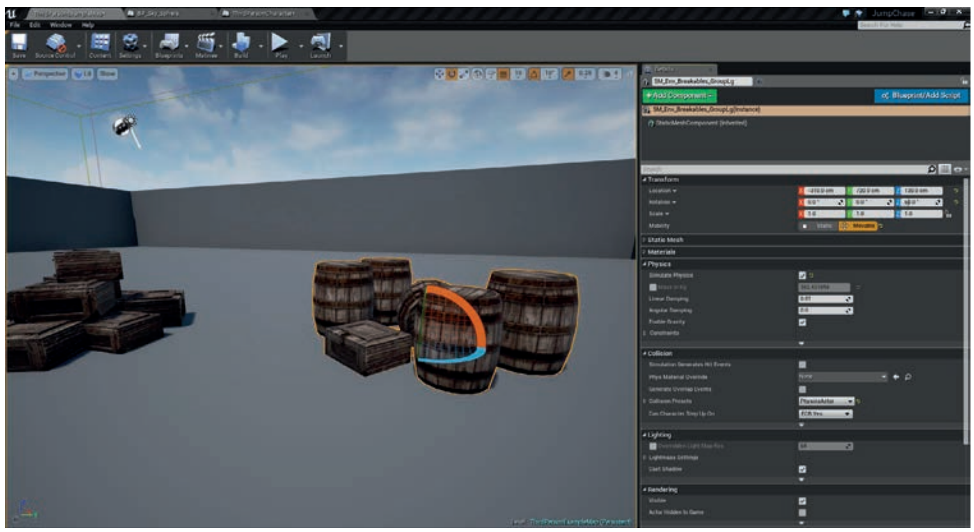


Figure 14.19: The barrels have got physics activated in the game engine

The screenshot above (Figure 14.19) shows how a games engine applies physics, with a simple tick box and then the application of various different settings, such as how much mass the object has. The image also shows the three different vector values that are applied to the position, rotation and scale of the object. These vectors all contain three values which correspond to the *x*, *y* and *z* coordinates of 3D space. The object is selected using the rotate tool and each of the three axes are presented as a direction in which it can be rotated.

The application of maths while the game is running is dependent on the features of the game. One common maths function that is applied to game objects is called lerp, which stands for linear interpolation. This is movement from one position to another and is used on moving platforms, simple enemies and health bars, for example.

Prototypes and engines

If the development is taking place in a games engine, the correct choice of engine must be made. There are many to choose from and, as most of them include a purchase price or payment of royalties, it is important to get the choice correct from the start. Two of the most popular engines, Unity® software and Unreal Engine, have

achieved their success by being able to port games onto multiple platforms, whereas other engines such as Cocos2D-x™ are aimed at one particular platform.

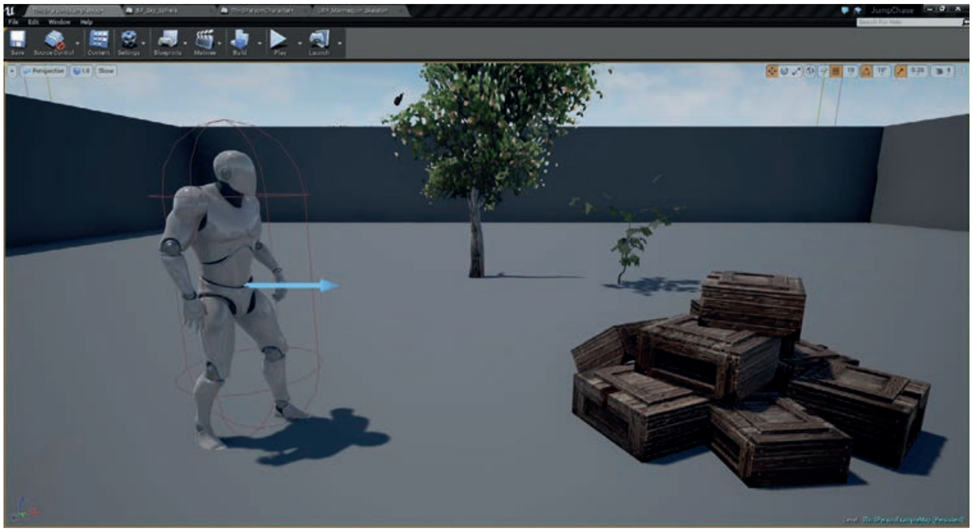


Figure 14.20: Game mechanics should be tested in small prototype levels

Once the games engine has been decided upon, a number of rapid prototypes will be created to test and show off some of the game's unique features. These prototypes will be used to check the feasibility of a feature, explore the timescales needed for development or to solve problems and to identify issues early on. The prototypes will sometimes be used to create trailers or promotional materials for the game, but developers need to be careful because players have long memories and if the final product seems too different from early footage then complaints will be likely.

Tools and techniques for development

There are many different tools and techniques used in games development, and the more mature the industry becomes the more creative developers are getting.

The image below (Figure 14.21) shows landscape sculpting, which is a technique available in many games engines whereby a large flat mesh is created and the developers are given various sculpting tools in order to create a realistic landscape.

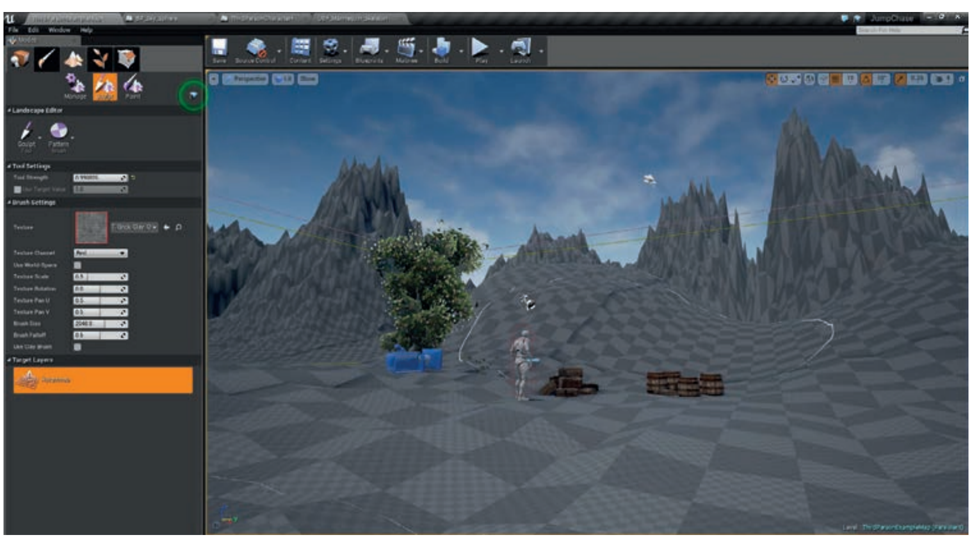


Figure 14.21: Landscape sculpting can save huge amounts of development time

A multi-layered landscape material will then be applied which allows the environment designers to colour in the landscape with different textures that will represent, for example, rock, sand and grass (see Figure 14.21).

Another technique is high-to-low poly modelling whereby a 3D modeller will create a highly detailed version of a mesh and then create a texture from the model that is then applied to a low-poly mesh. Doing this creates the impression of detail without having to render complex shapes.

Choosing the right tools and techniques depends on the type of game in development but also the experience of the development team.

Quality assurance

The quality assurance (QA) process begins during the game design with the designing of the test plans, and as development begins the testing team will start to grow. The testing process is not just about finding problems – it is also about suggesting potential fixes and refinements. Testers are more commonly known as QA technicians and it is a very responsible role that requires people with meticulous attention to detail. Testers should not just be people who enjoy playing games, but people who are able to think of all of the different ways that a game can be broken, and are able to follow formal processes to document every problem that they find in a way that can be understood by the programmers.

The testing is split into different phases and begins with unit testing all of the individual technical elements of the game. This is followed by alpha testing the first complete build, beta testing the next release after bugs have been fixed and then a final quality test to ensure that everything is working. Sometimes the developers will offer a beta build to the public for testing, as this gives them the chance to get much more feedback.

Technical constraints

While we have already looked at the design constraints that can affect how the game will look and feel, a number of technical constraints may affect the development stage. The game will have a set budget and this will dictate how much time can be spent on development, as most of the team will be on fixed-term, temporary contracts. Players often ask why games do not have all the features it is possible to have and budget constraints are one of the biggest reasons for features being cut back.

The size of the servers will affect the development of an online multiplayer game. The number of players who can access the game online at one time is known as the maximum concurrent players. If the server is unable to cope with all of the potential players across the world, then the workload may be split between multiple servers for set areas such as, for example, Europe and North America.

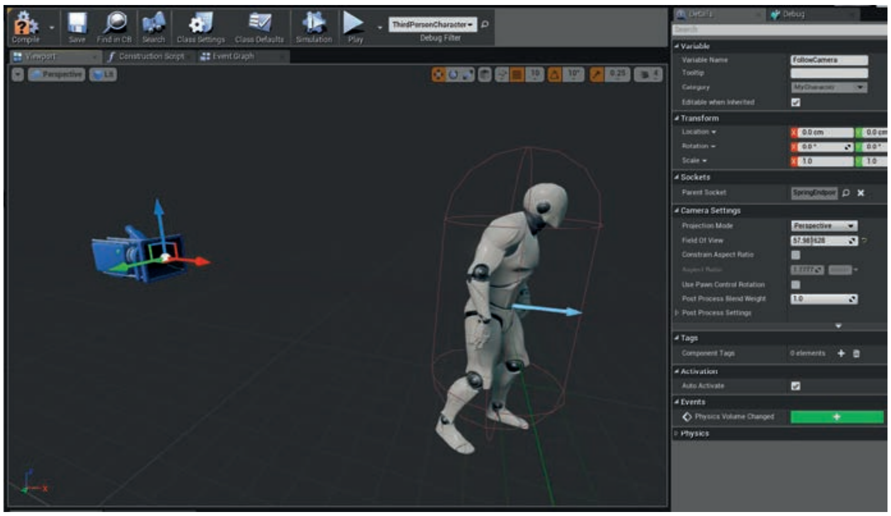
Developing computer games

In this section, we will follow the step-by-step development of a third-person game called Jump Chase where the player has to race through different levels to get to the end goal within a set time limit. Known as a platformer or platform game, this style of game has always been a popular design.

Worked example: Creating Jump Chase

Visual style

As a platform game relies on precise jumps and timing, the perspective works best with a third-person avatar so that the player can judge all their jumps carefully. The game is being made in Unreal Engine 4 and, as the screenshot below shows, the sample third-person character template contains an animated character mesh which is followed by a camera. That camera has its own settings and this includes field of view, or area of vision, which is the extent of the area that the camera captures. A wide field of view will let the viewer see more of the scene but this may scale down automatically if the display being used does not have enough resolution to support it.



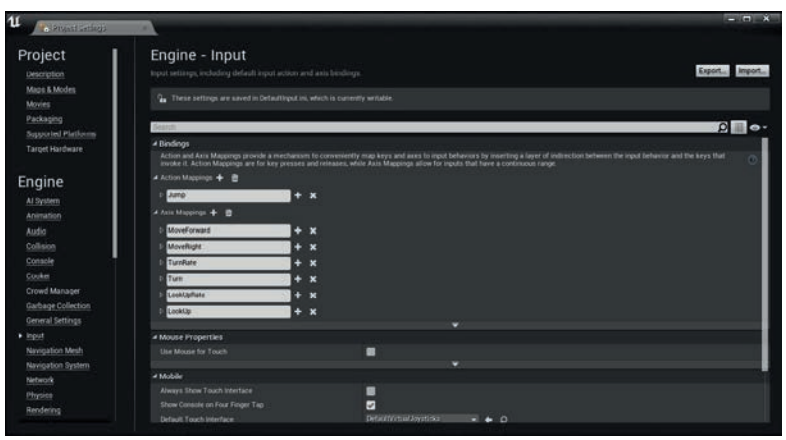
Link

For more about avatars and omnipresence see the Interaction model section.

▶ **Figure 14.22:** The field of view is set by the camera that follows the player's character in this third-person game

Input methods

In Jump Chase, the player needs to be able to control the main character as easily as possible and there are a number of input methods available for this, depending on which controller the game will use. This game is played on PC, which means that it could be controlled by a keyboard/mouse or by a gamepad.



▶ **Figure 14.23:** Player input can be mapped to keyboard, controllers or even both

The default settings for the third-person template in Unreal Engine supports mouse and keyboard but it would be relatively easy to add gamepad support as well. While many different gamepads can be purchased for PC, the majority of them tend to have the same configuration so that they will work with the bulk of PC games. The code for the character will not specify a particular key or button but instead it will be named, for example MoveForward, Jump, LookUp. These labels

are then mapped to one or more keys, so the walking movement may be mapped to the W, A, S and D keys. Some games allow menu options to provide customisable keys so that the player can choose exactly which keys they want to use. Some players may prefer the keys to be spread out, but some prefer them closer together. It can also work better for left-handed players to move the controls to the I, J, L and M keys. This feature is not usually available on console games, even though you can plug in a keyboard, but players are given control configuration choices in some games. Games consoles are sold with their own bespoke controllers and the designers will normally have used all of the button options available.

Asset integration

Once assets have been created for Jump Chase, they will need to be imported into the games engine. 3D modelling software exports meshes in a format called FBX that incorporates the mesh, textures and animations all in one file.

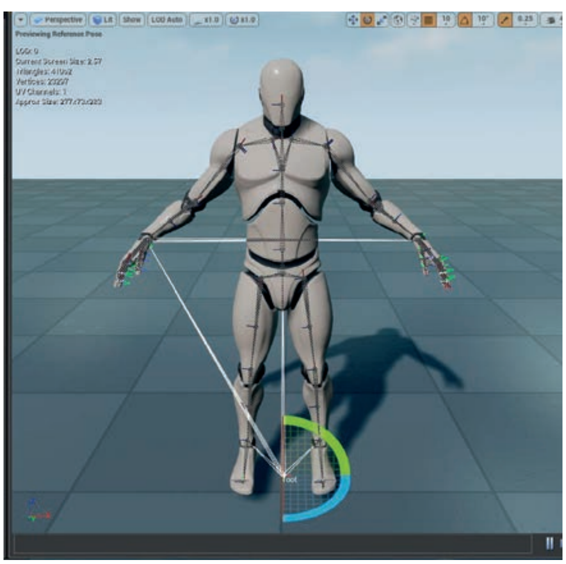
- Graphics: Graphics can be created in two different ways, using raster images or vector images. Raster images (known as **bitmaps**) are made up of a series of pixels and it is important when making games to keep the number of pixels in an image (its resolution) as low as possible without losing quality. This will make the game run more smoothly. A raster image with very few pixels will be a small file size but will have a very blocky style known as pixel-art style. The more pixels there are, the smoother the image. In Jump Chase the designers will need to make raster images for the textures applied to the 3D models and for the game's interface.

Key term

Bitmaps – images that are made up of individual pixels that together display a picture.

Vector graphics use points and lines to define an image. Vector graphics take more computational power to work with so are not used as frequently in games, and very few games engines support them. Even though vector graphics scale better, it takes less memory and processing to use a series of differently sized bitmaps.

- Texture mapping: All of the meshes that are imported into Jump Chase will have textures applied to them as set in the software in which they were created. One way to make a game run at a faster frame rate is to have different assets share the same texture file. This can be done when one asset, say, a floating platform, uses the top half of an image file for its metal texture and another asset uses the bottom half of the image. This means that two assets only need one image file. Textures are mapped to 3D meshes, which means that each flat surface on the mesh is tied to a particular part of an image file. If you think about a textured die, each face of the die would have a square in the image file and these squares would be mapped to each polygon in the mesh.
- Animation and video: The main character in Jump Chase is going to have different animation **cycles**. The default character has an **idle cycle**, a jump cycle, a walk cycle and a run cycle. The state machine uses the idle cycle as its **default setting** and, when the jump button is pressed, it will transition into the jump cycle.



Key terms

Cycle – instructions or operations which are repeated.

Idle cycle – the animation that a character displays when the player is not moving them. A character in an idle cycle may stand still and breathe or tap their foot after a few seconds.

Default setting – the setting selected by the games engine when nothing else has been input by the developer or, if the game is running, by the player.

Figure 14.24: Game characters can be rigged with bones to allow them to contain multiple animations

The walk and run cycles are blended together so that, when a speed value stored on the Jump Chase character changes (when the player pushes or releases one of the movement buttons), it will transition from idle to walk and then run, depending on how long the movement button has been pressed. All of these animations are used with the skeletal rig that is inside the mesh. Figure 14.24 shows what that rig looks like and how the games engine can be used to manually create new animations, if needed.

Other animations may be used in Jump Chase such as cut scenes or scripted events that move the story forward. Cut scenes will usually be recorded animations that are imported into the engine as a video file or, similar to scripted events, a series of animations controlled by the games engine. Unreal Engine uses an animation system called Matinee that allows the developer to use existing assets and levels to create animated sequences, which are observed by cameras in the level. Therefore the player will lose control of the character while the cut scene or scripted event takes place.

- Audio: Audio is crucial to the experience of playing a game. The virtual world is made much richer by music and sound effects that create memories and by educating the player in how the world works. As for graphic assets, sounds can be imported into the games engine after they have been created in other software such as Logic®, Audacity® or GarageBand®. The level in Jump Chase will have fast-paced, frantic music that indicates that there is a time limit, but there will also be a constant ticking sound until the time runs out.

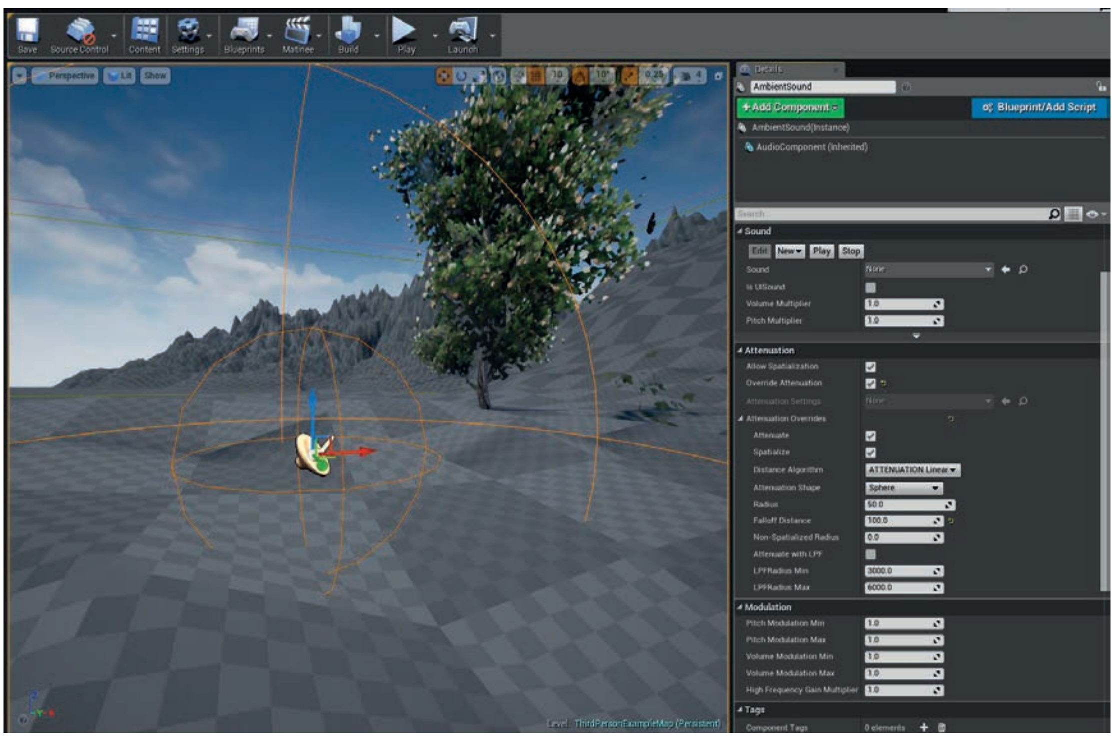


Figure 14.25: Audio can have 3D properties so that it can become louder the closer you get to an area

Sound assets can be 2D or 3D. 2D sound assets are heard by the player at its volume setting and cued to play as programmed by the developer. Sounds can use triggers, so when the player steps into a certain area, such as a creepy forest, they walk through an invisible trigger and a sound effect will play (the cry of an owl, for example).

3D sounds work in the same way but they are location specific, so as the player approaches the sound will get louder and as they move past it the sound volume will diminish. In Jump Chase, there will be a creepy building and the sound of someone laughing coming from inside a room that cannot be opened, just to add a sense of horror to the level and to keep the player motivated to move forward. The sound clips have to be synchronised to the visual displays so that they happen at the right time. This is especially important when sound is being used for spoken dialogue as the character mesh has to move its mouth at the same time.

Artificial intelligence

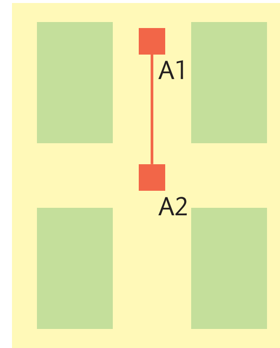


Figure 14.26: Grids are used in games to map levels for AI characters to traverse

Now that Jump Chase has some assets in place, this is a good point to start prototyping some of the more advanced features to ensure that they work correctly. The game features some roaming robots that will move around the ground and make sure that if the player falls down then there is still a challenge as they try to get back onto the platforms above.

The roaming robots will use a path-finding algorithm, a search algorithm, that needs to be tested in a prototype. The robots will consider the landscape to be a **grid** that is divided into separate **nodes**.

The code needs to be written such that the robots move between two different points. We need a way of working out the best way to move from one space to another and the breadth first search (BFS) algorithm can be used. BFS is an algorithm that shows the shortest route between objects on a grid. It does this by taking one step at a time and finding links between the different grid nodes. This is similar to a theory called six degrees of separation that says everything is six or fewer steps away, so that any two people can be connected, through a common experience such as being in the same film or going to the same school, by a maximum of six steps. BFS creates a queue of all of the nodes that are connected to the current node and then loops through them, one at a time, to see if they are connected or have been explored.

In C++, the programming language that the Unreal Engine uses, the BFS algorithm, would look like this.

```
//Loop until node queue is empty
while (!nodeQueue.isEmpty()) {
        //remove node ID from queue
        int visitedNode = nodeQueue.removeFromQueue();
        //check all connected nodes in a loop
        for (int newNode = 1; newNode <= n; ++newNode)
        {
          //check to see if the newNode is connected and if it has been
          visited
          if (isConnected(visitedNode , newNode) ) {
                    if(!hasBeenVisited[newNode ]){
                              //add to queue
                              nodeQueue.addToQueueEnd(newNode);
                              markAsVisited(newNode);
                    }
          }
        }
}
```

Key terms

Grid – a network of lines that is projected onto a game level in order to split it into logical sections.

Node – a point where lines intersect.

Link

For more about path-finding search algorithms, see the section Search algorithms.

This code snippet shows a section of the BFS algorithm where it is checking through all of the nodes connected to the current node and seeing if they have been visited yet. The robots in Jump Chase will do this before they move. Once the code loop has finished and created a list of nodes to move through, the robot will then make its moves.

More advanced features

- 3D rendering: Jump Chase has a 3D environment with sculpted landscape and audio. This is created in Unreal Engine and the platforms will be 3D meshes created in Autodesk® 3D Studio Max.

- Save game states and player progression: Another advanced feature is creating save files or auto-save points. Many modern games auto-save all the time, especially online titles. Third-person platform games like Jump Chase tend to use checkpoints, parts of the level that once reached are automatically returned to if the player's character loses a life. A checkpoint location will be stored in a save file locally on the player's system or on a server if it is an online game.

  Local file saves are also used to save progression information in Jump Chase. Details about how far the players have reached and how quickly they completed the challenges are added to leaderboards. The game will also use an achievement system so that when a player achieves certain goals in the game, such as completing it on the hardest level of difficulty, they will be able to see a trophy on the achievement screen.
- Multiple players: Games developers have to think very carefully about any multiplayer or networked features they are going to put into a game. Multiplayer games contain many advanced features such as player matching, which is a system that finds different people around the world who are of a similar level to you and will provide an appropriate challenge for you. Players do not like to be pitted against opponents who are too strong or too weak for them as this makes the experience unsatisfying.

  Ensuring that network connections are maintained is important, but this is not something that is in the developer's control. Game publishers will be responsible for creating servers that online games connect to, and the reliability and availability of these servers is dependent on how much money is spent on them. The servers have an important function in allowing the player's systems to communicate with each other and in maintaining any online leaderboards and achievements.

This example is designed to give an overview of a developed game but is too complicated to be created by a single learner or a small team. The scope of this unit is to provide an understanding of the whole process and to create a simple working game that will provide an understanding of how to create something that could be developed into a wider, working title.

Testing computer games

Once the development is completed to a point where testing can begin, it is important to get started as the more time that can be given to testing, the better the final product will be. The focus of the testing is split into four areas that look to answer the following questions.

- Playability – do all the features in the game work? Can the player do everything that they should be able to do? Is it possible for the player to get stuck anywhere?
- Compatibility – does the game work on its intended platform? Does it load and save appropriately? Do all the system's controllers work with the game?
- Stability – can the game be played from start to finish without breaking? Is the frame rate consistent throughout?
- Acceptance – do the players enjoy the game? Is the challenge at the right level?

Testing tools

Games engines and IDEs contain tools that help with the testing phase of games development. Figure 14.27 shows a statistics window that can show details of all of the assets in the scene and how they are being rendered.

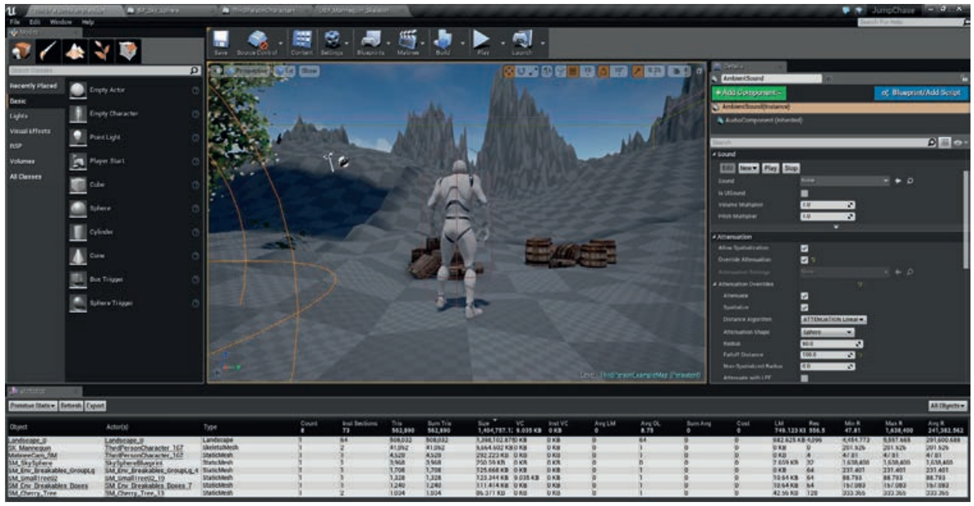


Figure 14.27: A statistics window

Key term

Bug – a problem in code that causes errors or glitches in a game.

Other tools include debugging tools like breakpoints and watch values that allow the programmers to automatically stop the game at certain lines of code within a function to check to see which values are being stored. This will be useful if a particular game feature is not working correctly because it will help the programmer or tester to solve the problem (or **bug**) by processes of elimination. When the programmer knows where the game is going wrong, they can figure out how to fix it.

Feedback

Feedback from testers, clients and players is very important and this is why end user testing or user acceptance testing is crucial. Testing by players may be done by small focus groups at the development studio, in larger sessions at games conventions or a game may be opened up to the public as part of an open beta. It is common for small testing groups to have to sign non-disclosure agreements (NDAs), which are contracts that prevent them from revealing details about the game. This way, if anything is taken out of the final version of a game, the public will never know about it.

End user testers will be asked to consider the following.

- Effectiveness – how well does the game play? How good are the controls?
- Presentation – does the game look good? Does the art style work well?
- Performance – does the game run smoothly? Has it lagged at any points?
- Accessibility – how easy is the game to control? Is the difficulty level manageable?
- Portability – does the game work well on different platforms? Does it work well on different screen sizes?
- Robustness – can you break the game? Can you get stuck anywhere?
- Purpose – do you understand what you are meant to be doing in the game? Do you understand why you are doing it?

Refinements

The feedback gained from users will go into making refinements to the game, but the scale of refinement depends on how much time is left before the final deadline and may also be dependent on the budget. Small changes to layout and challenge level can be considered but anything that requires new meshes or characters will be impossible

after the beta stage of development. Some console games developers, when criticised on the performance of their games, have blamed the differences between console development kits and the retail versions of the consoles that players have at home. The disparity can be because the retail versions are updated more frequently than the console development kits in order to solve security issues. But if the console development kits do not have these updates, then developers are essentially having to test their games on slightly different systems which can lead to problems.

Theory into practice

Skills, knowledge and behaviours

If you want to work in the computer games industry, you need to develop the following skills.

Planning and recording

- You need to be prepared for the massive workload that is involved in the games development process. You must have skills that include the ability to set your own targets, consider timescales and decide how you are going to get feedback from colleagues.

Reviewing your own work

- You must be adept at reviewing your own work and be willing to admit when something is not as good as it should be. Teamwork is not the place for ego and if you cannot put your hands up and ask for help when something is not going to plan, then working in creative teams such as games development studios is not the right place for you.

Responding to feedback from others

- You must be able to collect useful feedback from other creative professionals and end users, which means putting your product in the hands of critics and being able to listen to both positive and negative feedback.
- You must be able to look at the feedback that you receive and compare it against the original requirements of the brief. If an end user is demanding unrealistic functionality, then their feedback is not that useful, but if someone is in your target audience and they are not enjoying the game, you must find out why and try to see if you can do something to improve their experience.

Reviewing computer games

After a game has been tested, it will then be reviewed. The review of a game is usually completed by the development team and the games publisher. It could be tempting to skip the review stage if deadlines are pressing but it is a really important step as it is an opportunity to evaluate the entire design and development process.

Quality

The reviewer will first comment on the quality of the game. How well is it made? They will consider the textures, the animations, the game mechanics, everything. Players tend to pay around £40 for a console or PC game or half that for a digital download game for PC. They expect to get a quality product for their money. This means that all the controls need to work well with the game, the graphics need to look good and the challenge levels need to be appropriate for the intended audience. All of these aspects will be checked by the reviewer.

Suitability for audience and purpose

The reviewer will consider who the audience is for the new game. They will make comments about whether or not the game has been designed with the audience in mind and to what extent the audience will be satisfied by the title. It is not just a question of age here – it could be a focus on the ability of the players, their intelligence or their history with that franchise of games. If a sequel comes out that does not refer back to the original story or move the story forward, then it is not considering its audience, it is just trying to make more money without giving the audience what it wants and expects from a sequel. The game must also be reviewed against the original purpose intended for the game.

Original requirements

The reviewer will also look at the original requirements of the game. They will look back to the original high-concept design document and comment on whether or not the game has met the outcomes that it originally set out to achieve. It may be that changes and refinements during development have changed some things, but a game should generally end up in a similar state to how it was originally conceived and designed.

Legal and ethical constraints

The games publishers should get a legal team to review the game and its content if there is any chance that it breaches any legal constraints such as copyright law or ethical considerations. Smaller development studios will not have the luxury of their own legal team and may have to check potential legal issues themselves or pay for legal advice from an external firm.

Technology constraints

Once the game has been made, its technical requirements will be confirmed and the reviewers will be able to check to see if these cause any problems. In the previous generation of consoles, the Xbox 360® used DVD discs and if a game was too big it would spill onto a second disc, which meant more expensive boxes were needed and players would be frustrated at having to change game disc halfway through the game. PC gamers pay very close attention to the recommended technical specifications so that they can play their games without having to sacrifice graphics or quality. When the quality team reviews the game, they will set out the minimum technical requirements for the game and the recommended requirements, as they will be advertised.

Strengths and improvements

A game is always sold on its strengths, and the reviewers will make sure that they are fully aware of everything that the game does well and anything that is not perfect. If a game is found to have an issue and it is too late to make fixes or refinements, then the developers must produce a **patch**. Games which have been shipped by the time problems are found will have what is called a 'Day 1' patch. A 'Day 1' will automatically download and install the fix when the player goes to play the game for the first time, but only if the player's system has internet access and the game knows to check for patches whenever it loads. Patches can continue to be delivered to the player's system long after the game has been released, but the longer the delay between fixes, the more dissatisfied the player will be.

Key term

Patch – a series of code fixes that are downloaded and applied to the game code in order to fix problems.

Platforms and compatibility

The review team will look at all the different platforms that a game has been developed for and ensure that it works correctly and runs effectively on those platforms. There may be a certification testing phase where game code is sent to console manufacturers or digital distribution platforms to ensure that the game's install process works properly and that it makes proper use of any system features that it needs, such as network access or game saves.

External reviews and quality characteristics

Video games are big business, with millions of pounds a year being spent on games development and even more being spent on research into new hardware systems and advances in technology. Games journalism is crucial to both players and developers because the review scores that games are given before they are released can make or break it. While the games magazine industry is starting to fade, games websites are more popular than ever and players will often look to IGN.com, Kotaku.co.uk and Gamespot.com to decide what their next purchase is going to be.

Some games reviewers have got into trouble in the past for giving positive reviews to highly flawed games and have been accused of being influenced by games publishers through being given trips and lavish gifts. This has led to an increase in popularity of Metacritic.com, a site that gives a single score for a game based on the average of the scores from many different review websites. It also allows users to give a score to games, so that the opinion of the audience, as well as that of the critics, can be seen.

Further reading and resources

Game software to download for free

Unreal Engine 4 – **www.epicgames.com** (free for personal and educational use).

Unity® 5 – **www.unity3d.com** (free for personal use, educational fees vary).

Autodesk® 3D Studio Max – **www.autodesk.com/education** (3 year licences for free for students and educators).

Tutorials

Brackeys.com – brackeys.com excellent free tutorials.

Pluralsight Creative – **www.pluralsight.com** subscription-based video tutorials.

Articles

Richard Bartle – 'Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs'. **http://mud.co.uk/richard/hcds.htm**.

David Keirsey – 'Please Understand Me II' (1998); **www.keirsey.com/**.

Assessment practice 14.2

B.P3 B.P4 B.M2 C.P5 C.P6 C.P7 C.M3 BC.D2 BC.D3

Now that you have shown your knowledge to the games publisher about different games platforms, you have been asked to design a game. The publisher wants you to create a game for the web using the recent WebGL plugins that are available on the main games engines.

Remember that WebGL allows developers to create whole 2D and 3D games that will run in a web browser.

In your design, you must show that you understand the constraints of designing for a web platform and that nothing is asking too much of the system.

The game's design is your choice. The publishers are open to seeing your ideas but they have specified the following requirements:

- a PEGI rating of 7 or lower
- either a 2D or a 3D design
- a third-person player character.

You should produce a GDD document and a set of design documents that justify your design decisions, showing how they fulfil the purpose of your game and meet the client's requirements. You will need to review the designs with others to make refinements.

You might create the following:

- storyboards
- level design
- a summary of game mechanics
- game screen mock ups.

Having completed the design for your WebGL game, it is now time to create it. You have been asked to produce two working levels for the game that you have designed with a welcome menu and a game over screen.

Remember that you have learned how long and complicated the game development process can be. Do not be too ambitious in your design. You can achieve high grades with a simple game design that works well.

You must ensure that the game is optimised for web delivery, which means that textures should be minimal resolution, any 3D meshes should be low poly and any audio should be kept to a minimum. Keep a production diary of your development and testing which records your organisation and self-management.

After you have created the game levels, you must test it fully and produce a review for the clients that evaluates the design.

Plan

- What am I being asked to do?
- What sort of game would the audience want?
- What resources do I need to complete the task?

Do

- Have I got a full set of design documents?
- Have I thought about everything that needs to be designed?
- Can I identify when I have gone wrong in my game production and get myself back on course?
- Am I recording my own observations and thoughts in a production diary?

Review

- I can consider all constraints involved in a web game.
- I realise where I might not fully appreciate how many assets I need to create to make the game look professional.
- I can explain the skills that I employed and the new ones which I have developed.
- I can explain what success looks like.

THINK FUTURE

Gemma Kellner

Games programming university student

I'm an undergraduate student on a Games Programming degree course at university. I'm currently in my second year and, so far, I have learned how to write games code in different scripting languages, use 3D modelling software and improve my maths so that I can code more efficiently. My course is a lot of hard work and I know that when I finish I am going to have to make myself really stand out in order to get a job in a very competitive industry. I spend my free time working on my art skills, so that when I am sent art assets I have a better understanding of how they will be used in the game and affected by the code that I write. I'm studying hard to ensure that I understand the entire workflow of games development. I will also be entering Game Jam events to network and improve my technical skills, such as writing code that will create the game's rules, the player controls and the win conditions.

Focusing your skills

Planning a game's development

It is important to consider the platform and audience for any game that you plan to make.

- Consider the timescales – how long have you got to complete your project?
- Have you done any market research? Is your idea that unique?
- Why is it fun? Would people be constantly stimulated playing your game?
- What is visually appealing about your game?
- Who is the game aimed at?
- How long would people need to play for? Is that amount of time realistic?

Working in a professional games studio

Professionalism

- To be professional is to understand that nothing is personal and that everyone is working together for the combined good of the game. You must put player satisfaction first and be willing to put the extra time in to ensure that this happens. You need to demonstrate your professionalism by being truly expert in your specialist area and delivering what you promised. You have an individual responsibility to do your job in a thorough and timely manner.
- Communication skills are key to being a good professional. You must use face-to-face meetings, phone calls and emails in appropriate situations, and make sure that your team and clients feel comfortable at all times. You must demonstrate good etiquette to colleagues, always be polite and never be overly personal.
- You may need to demonstrate leadership skills. Taking on a leadership role makes you even more responsible for your product. If you are a leader, then you are accountable for the team that you are leading and it is up to you to organise and motivate them to get the job done.

Evaluating outcomes

- You must be able to evaluate how well the product has met the original requirements of the client brief. You may be asked to make any recommendations about taking the project forward or decisions about last minute patches or refinements. Your recommendations must be based on considered review and reflection.

Evaluating targets

- You must be able to evaluate the targets that you set yourself or that your team leader set. Did you meet your targets? Was there enough time? Would you have approached a problem differently given more time? You must be able to ask yourself these kinds of questions to obtain insights into your own performance.

Getting ready for assessment

Rupert is working towards a BTEC National in Computing. He was given an assignment that asked him to write a blog about 'Social Trends in Gaming' for learning aim A. He had to cover all the different areas of gaming trends in order to explore the changes in recent years and make some comments about where gaming could go, based on emerging technologies.

Rupert shares his experience below.

How I got started

I made a list of all of the different areas I needed to cover, from my class notes, and created individual blog pages for each of the headings. My headings were: Genres, Players, Production, Multiplayer, AI, Emerging Tech and Security.

It made me feel more organised to have all of the pages ready before I started, even though lots of blank pages were a bit daunting. I made sure that I had tagged the pages with the titles too, so that people searching through my blog would be able to find the information they needed quickly. I then went through all the notes I had made in class and wrote the appropriate tags on the top of the pages. For example, when I had made notes about the Oculus Rift™ being good for first-person shooters, I wrote 'emerging tech' and 'genres' on top of the pages. This meant that I could order my notes ready for writing them up in the blog. Some of the note pages had multiple tags written on them and I photocopied these so that I could keep separate stacks of notes for each blog section.

How I brought it all together

I then typed up my notes into the different sections of the blog but I made sure that I wrote it in a more structured way than the notes. My notes were originally just for me, so I had to make sure that they made sense to anyone reading them. A blog does not have to be written formally, like an essay, but it should be grammatically correct and make sense.

Then I found lots of images that supported what I was writing about and put them into each section, taking care to arrange them so that the pages flowed nicely. I kept notes of where I got my images from and referenced them in my blog.

I finished each page with URLs linking the reader to websites where they could find more information.

What I learned from the experience

I learned that I can write a lot from keeping good notes in class, from lectures and discussions. My handwriting is bad, so some of my notes were stored on my phone because our tutor trusts us to make notes that way. I was able to cut and paste these notes because they were already text. I think that in future I will keep all my notes this way and I might save up for a tablet, as it would be easier and quicker to type on.

I also learned that there is a lot of information on the internet about games trends and not all of it is true. There are a lot of opinions online, and if you find out some information you should make sure that it is posted on a few different sites to check that it is likely to be true.

Think about it

- Have you been keeping notes in classes so that you can refer back to them when writing assignments?
- Do you have a list of reputable games-related websites that you can trust to give you reliable information?
- Have you been keeping a note of all the places you got information from so that you can reference it?

Website Development 15

Getting to know your unit

Assessment

You will be assessed by a series of assignments set by your tutor.

The internet has grown and developed so much within the last decade, enabling new ways of communicating and sharing information. It has transformed the way in which people use information technology. Millions of web pages are created daily so it is important that they are engaging and capture the interest of the public. Website developers need to understand how to resolve complex problems and develop innovative solutions.

The evolution of the worldwide web has brought about more exciting opportunities for businesses and individuals. Businesses can now sell goods and services online, which extends their customer base, potentially worldwide. Individuals can communicate affordably with people anywhere in the world. Never before have there been so many opportunities for people who use the worldwide web. It is now a place where people can produce their own blogs and even make substantial amounts of money doing so.

As the worldwide web evolves, information technology professionals must evolve with it. In this unit you will learn the fundamental principles and key concepts of website design and development. The skills you acquire will benefit you if you wish to further your studies in higher education or prepare for employment in the website development sector.

How you will be assessed

This unit will be assessed by a series of internally assessed tasks set by your tutor. Throughout this unit, you will find activities that will help you work towards your assessment. Completing these activities will not mean that you have achieved a particular grade, but you will have carried out useful research and preparation that will be relevant when it comes to your final assignment.

In order to complete the tasks in your assignment successfully, it is important to check that you have met all the Pass grading criteria. You can do this as you work your way through the assignment.

If you are hoping to gain a Merit or Distinction, you should make sure that you present the information in your assignment in the required style. For example, Merit criteria require you to analyse, discuss and optimize, and Distinction criteria require you to assess and evaluate.

The assignment set by your tutor will consist of a number of tasks designed to meet the criteria in the table. This is likely to consist of a written assignment but may also include activities such as the following.

- Explaining how websites are used and the principles that underpin them.
- Using different design techniques to provide innovative solutions to complex problems.
- Creating a website based upon your design documentation.

Assessment criteria		
This table shows what you must do in order to achieve a **Pass**, **Merit** or **Distinction** grade, and where you can find activities to help you.		
Pass	**Merit**	**Distinction**
Learning aim A Understand the principles of website development		
A.P1 Compare the principles of website design used in two websites, including their suitability for the intended audience and intended purpose. **Assessment practice 15.1**	**A.M1** Analyse how the principles of website design are used to produce creative, high performance websites which meet client requirements. **Assessment practice 15.1**	**A.D1** Evaluate how the principles of website design are used to produce creative, high performance websites which meet client requirements. **Assessment practice 15.1**
Learning aim B Design a website to meet client requirements		
B.P2 Produce designs for a website that meets client requirements. **Assessment practice 15.2**	**B.M2** Justify the design decisions, explaining how they will meet the users' needs and be fit for purpose. **Assessment practice 15.2**	
B.P3 Review the website design proposals with others to identify and inform improvements. **Assessment practice 15.2**		
Learning aim C Develop a website to meet client requirements		
C.P4 Produce a website for an intended audience and purpose. **Assessment practice 15.2**	**C.M3** Optimise a website to meet client requirements. **Assessment practice 15.2**	**BC.D2** Evaluate the design and optimised website against client requirements. **Assessment practice 15.2**
C.P5 Test the website for functionality, compatibility and usability. **Assessment practice 15.2**		**BC.D3** Demonstrate individual responsibility, creativity and effective self-management in the design, development and review of a website. **Assessment practice 15.2**
C.P6 Review the extent to which the website meets client requirements. **Assessment practice 15.2**		

Getting started

Think about some of your favourite websites. What is it about them that makes them your favourite? Is it the colour, the positioning of elements, or the font type used? Working in small groups, list some of your favourite websites and note down what you think it is that makes them stand out from the rest.

A Understand the principles of website development

Websites have changed tremendously throughout the years. When the web was first developed, it was used by scientists in research institutes to share ideas. Early websites included **hypertext** and hyperlinks as a way of moving through all the text. In today's world, websites are still used to share ideas, but the web has evolved into something essential to our way of life.

Key term

Hypertext – text that contains links to other bits of text.

Purpose and principles of website products

Websites are now so much more than simple links that provide information. Consider a very basic news website. It is full of up-to-date content such as videos for streaming, polls where visitors can publish their thoughts, and even basic games. Websites and the businesses that run them are now in competition against one another to get you to view their products. Therefore, website developers need to understand the purpose and principles of website design so that they can make websites which are fit for purpose but also stand out in people's minds for the right reasons.

Purpose of websites

Websites have different purposes and perform different functions. Traditionally, websites were used to provide information. However, content-based technology called Web 2.0 has been developed to enable users to interact and add their own content, knowledge and opinions. Web 2.0 technology has enabled the development of the following.

- Wikis – places where all users can contribute to information, the biggest example of which is Wikipedia, an encyclopaedia website where anyone can contribute to any article and even create new ones.

Theory into practice

In this unit you will come across many technical terms. Go online and develop your own wiki that will serve as a glossary for all the terms you come across within this unit. This can be shared with your class and you could all contribute to it.

- Blogs – online journals that include tools for readers to comment and contribute. Anyone can become a blogger and create a blog, most of which are public (although there are some private ones).
- Social networking – websites such as Facebook Inc, LinkedIn® and Twitter™ that allow people to communicate by creating a profile. These social media websites provide tools that enable users to correspond by posting status updates or tweets. They can also have private conversations and play games. Users are encouraged to make 'friends' with other users, although most people using Facebook Inc are friends with people they know in real life and friends of friends online. On Twitter™, you follow celebrities and well-known figures as well as friends, but the celebrities are unlikely to follow you. There are other social media websites that offer more specific services, such as Flickr™, which allow users to share and rate photographs.
- Online applications – online applications allow users to use programs via the internet, rather than purchasing a program and installing it on a local computer. This means that the applications are accessible anywhere and are often free or have a small subscription fee. Google Docs™ is a good example of an online application.
- Podcasting – a way of making audio or video files available on the internet that can either be listened to or viewed. A podcast is saved to either a PC or a mobile device where a user can listen to or view the content while they are on the move. A podcast will be treated as a sound file (audio podcasts) or a film (video podcasts are also known as vodcasts).

Product and/or service-based websites

Table 15.1: Examples of product- and/or service-based websites

Service	Example types	Target audience	Key benefits
Commerce	• eBay • Amazon™ (Amazon and the Amazon logo are trademarks of Amazon.com, Inc. or its affiliates) • online banking	• consumers	Allows users to carry out transactions easily and conveniently on PCs and mobile devices.
Real-time information	• 24/7 news updates • travel information • weather reports	• information seekers	Provides immediate up-to-date information. For example, mobile phone apps can send notifications to inform you of recent news reports or train arrival times.
Communication	• blogs • Skype® • email • online gaming	• social networkers • information seekers • online gamers • adult and teenage consumers	You can receive immediate replies from anyone at no additional cost to your broadband or mobile fee.
Download services	• music downloads • film downloads • software downloads • gaming downloads • streaming services	• information seekers • entertainment seekers • online gamers • adult and teenage consumers	Flexibility and convenience of downloading different mediums to your PC or mobile device. Moreover, you can also now stream online content to your television, computer or mobile device.
Virtual learning environments (VLEs)	• Moodle • Pearson's ActiveLearn Digital Service • Blackboard Inc	• information seekers • consumers (teachers and learners)	Many benefits as learners have the flexibility of downloading information from the comfort of their own home. Moreover, learners can submit homework/assignments online or be set independent study tasks.

Requirements of websites

Discussion

Hackers have caused controversy by being able to hack into personal data stored on websites and then release the information to the public over the internet. This has included credit card details. In small groups, consider the benefits and risks of buying products and services online.

The following are key requirements of all websites.

- User-friendly – It is easy to use and understand. Understanding how to produce a user-friendly website will significantly improve the way in which visitors are able to understand and use your website in terms of layout, design and the content.
- Consistent – Consistency, that is, keeping things the same, is a key principle of website design. Consistency can apply to many different things within website design. For example, consistency in the colours and font typography you use, the positioning of links on pages and the style and images you use. An inconsistent approach to website design makes for a website that will ultimately frustrate users. Consistency makes for an intuitive and user-friendly website.
- Navigational – It is important that any website you develop is easy for users to navigate and find the information that they require. Think of a website's navigation like a road map to all the different areas and information contained within the website. A consistent navigation system will help your users to find the information they require more quickly. It is also important not to add in unnecessary levels of navigation.
- Customisable – New technologies have made it possible to customise website interfaces for website users. The benefit of this is that you can attract new users and keep existing users more engaged with your website. For example, consider the way in which some websites allow you to set your location on them. One advantage of doing this is that you can get regular news headlines about your specific area or receive local weather reports.
- Responsive – Responsive websites are those that are optimised to automatically adapt to the layout of the web browser you are using, whether on a PC or mobile device, without the need to resize or scroll excessively.

Research

Carry out some research into quantum computing. What implications will quantum computing have for the way we buy and sell products and services online?

Principles of website design

Millions of websites are built on a daily basis, so more emphasis is now given to the principles of good website design. Following these standard principles is important as it allows every website the opportunity to be successful in a market which is very competitive. When you combine these principles of good design with elements of innovation and creativity, your website has every chance of being successful. In terms of business, if you get the website right, the business is more likely to succeed.

- Usability – This is the ease of use of a website, that is, how user-friendly it is. For example, an important element of website usability is ensuring that website content is flexible and works on all browsers and devices.
- White space/spacing – This is the space on web pages that is left untouched and white. Website developers utilise white space to separate design elements such as text, graphics and other elements. Unless you utilise spacing, your web pages will look cluttered and messy, and it would be hard for visitors to interpret which words relate to which images and understand what they are looking at. Therefore, when designing a web page, it is crucial to use spacing between elements so as not to overwhelm and confuse website users.
- Site layout – This means giving consideration to where elements go on a web page and how content will be spread across a website. For example, where the header, navigation links, text and graphics will appear on the page. The layout used for one web page should be consistently used across all the pages on a website, where similar content is being included. For example, headers should be positioned in the same way on all pages. A good site layout is one that is uncomplicated, has clear navigation and is intuitive and user-friendly.
- Accessibility – Website developers must ensure that websites are correctly designed and developed to enable all users to have equal access to information. This is important as it removes barriers that would otherwise prevent a person with disabilities from using a website. The BBC is an example of a website that provides excellent website accessibility. It includes features that advise users on how to customise their computer set-up and remove barriers to using the website. People with poor vision can change the font size and colour. In addition, it provides accessible games for children with motor and cognitive disabilities. **Alexa Traffic Rank** consistently rates the BBC website in the top 100 websites worldwide. One reason for this is the excellent way that it caters for all users.

Key terms

Alexa Traffic Rank – ranks websites based on a combined measure of page views and the number of visitors. From this, it creates a list of 'top websites' averaged over three-month periods.

Serif – a type of font that has embellishments at the ends of letters. An example of a serif font is Times New Roman.

Sans serif – The word 'sans' is French, meaning 'without'. Therefore a sans serif font is one without embellishments at the ends of the letters. An example of a sans serif font is Arial.

Link

The World Wide Web Consortium is committed to ensuring that the web is accessible for all, regardless of their disability. Find out more about their Web Accessibility Initiative at **www.w3.org/WAI/**

- Navigation – It is important that users can navigate their way easily around a website. The navigation must be intuitive and simple to use. Good navigation on websites involves making it obvious where website links are and being consistent across the whole website. There should be an intuitive structure of pages.
- Typography – The type of font used on websites is important. Websites need to have clear and legible text in order to be user-friendly and accessible. There are two main types of font families: **serif** and **sans serif**. Most websites utilise a sans serif font such as Arial because they have no embellishments to the letters, making them clearer and easier to read.

Research

Research how typography is connected to disabilities such as dyslexia. When building a website, what font would you use and why? Consider what else you could do as a website developer to ensure that all users can access your website regardless of disability.

- Alignment – Alignment is how elements are positioned on a web page and how they interact with each other. For example, if you place an image on a web page which is aligned to the left and the supporting text was aligned to the right of the page, this would suggest to users that there is no relationship between the two elements, whereas, if they were both aligned left, there would appear to be a connection between the two elements.

- Clarity – Clarity, being clear, is key to good website design. It is pointless choosing a visually appealing design if users cannot understand it because it is not clear. Therefore, any website you develop must communicate what you want it to say clearly, but also have clarity of design and layout. For example, if you want a user to contact you, you should provide a visible email address or phone number. Asking users to fill in forms takes additional time and does not put the user first. Good design matters, but good communication matters more.
- Consistency – Consistency means keeping elements within your website the same throughout. This is considered one of the most important principles in website design. Website consistency involves using the same page layout, the same font and the same design and colour scheme, throughout. An inconsistent website is unintuitive and difficult to navigate.
- Accuracy – Everything included in the website needs to be accurate: that is, it should be correct and meaningful. To keep information accurate and current, websites should be updated regularly. Moreover, it is important that the information provided on websites is checked for factual errors as well as spelling and grammatical mistakes.
- Content – The content of the website includes the text, graphics (diagrams, artworks and photos) as well as media and objects. Content needs to be accurate, consistent in style (where appropriate), and you need to ensure that you have permission to use any content that you did not create from scratch.
- Media and objects – Some websites include interactive content, such as videos, background sounds, Flash images and **applets**. Media elements are used to captivate and grab the attention of website users. The use of media must be appropriate as its overuse can slow down a website, clutter the appearance of web pages and put off some users.

Key term

Applet – a mini software application that can be built into a web page, for example a calculator that can be used by website users upon clicking a button on a web page.

- Simplicity – Websites which are straightforward in their design are often the most visited. They manage to combine the content they need to get across with a clear and simple, easy-to-use design that attracts users back because they know they will get what they need from it without fuss. It is important when developing a website that you do not overcomplicate the design by introducing too many elements on the pages.

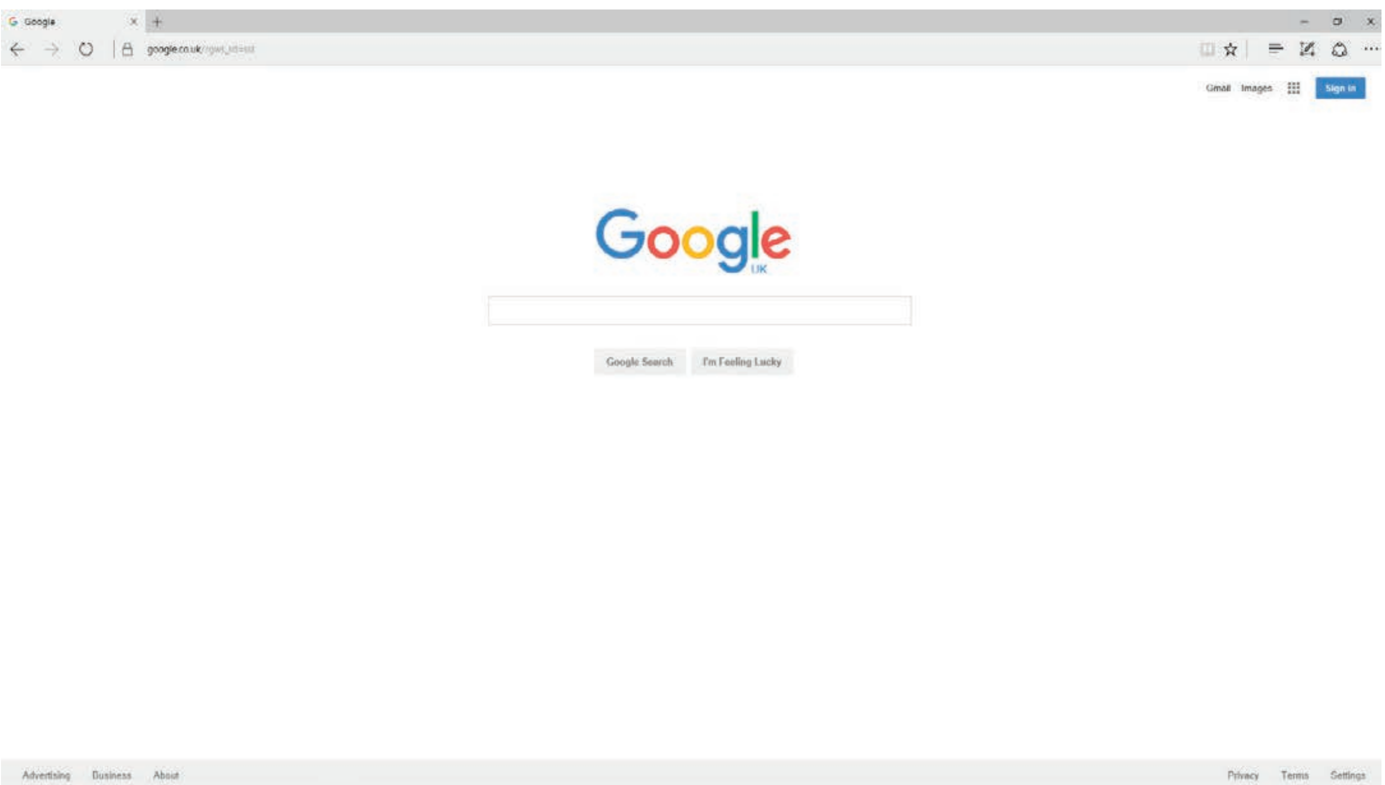


Figure 15.1: Google.com home page. (Google and the Google logo are registered trademarks of Google Inc., used with permission)

PAUSE POINT Alexa Traffic Rank ranks websites based on a combined measure of page views and the number of visitors. In August 2015, google.com (see Figure 15.1) was measured to be the most heavily visited website during that year. Consider what makes this website so appealing and popular. Print out screenshot(s) of google.com and annotate them in terms of how the principles of good website design apply to this website. Alternatively, if you do not wish to use google.com, consider another website that you think is good and identify where the principles are applied.

Hint Use the snipping tool within Windows (go to the Windows menu, All apps, Windows accessories, Snipping tool) to 'snip' parts of the website that adopt a principle of good website design. This is easier than printing screenshots and then cropping images.

Extend Consider additional principles of good website design that you believe are important. Explain what these principles are together with an annotated image to explain your reasoning.

Media and objects

Media and objects are used within websites to draw in and engage website users. As mentioned above, media includes videos, audio and interactive content. Moreover, objects such as applets, PDFs and Flash can be embedded into web pages. There are a number of factors you should consider when using media and objects.

- Position – Where should you put media or objects on a web page? As media and objects are used to grab the attention of users, they need to be placed somewhere where they can easily be seen, but you need to consider how prominently you want to position them.
- Colour – Colour is an integral part of website design, especially when you are embedding object elements such as Flash. Flash is a form of animation used to attract the attention of users. However, too many vibrant colours can be off-putting to users. You should also consider users who suffer from photosensitivity. Therefore, when using colours you should use them appropriately so that they are not overwhelming.
- Contrast – Contrast is used to differentiate between two or more elements on a web page. For example, if you were using a Flash image embedded in your web page, and the background used a key colour of the Flash image, it would be difficult for the user to see the Flash image clearly. Therefore you should use contrasting colours to differentiate the elements of a web page.
- Size – Screen size is a big concern when developing a web page. Media and objects are used to capture the attention of users. However, if you embed a video on the web page that takes up most of the screen, this will have implications on where the rest of your content will go, and users will have to scroll to see it. Therefore, sizing of media and object elements is important, and this should be thought through during the design phase. You will need to consider how the object will appear on a mobile. Will it work on a mobile?
- Appropriateness – When using media and objects, you must consider whether they really add anything to the effectiveness of your webpage. Although media and objects are useful to draw users' attention, the downside is that the inclusion of media and objects means that your web pages will take longer to load, especially on a mobile device and they might not be compatible with every browser. Therefore, you have to consider the appropriateness of using media and objects, and whether their inclusion will actually be beneficial to website users.

Creativity and innovation

With so many websites being developed on a daily basis, and businesses now heavily reliant on websites, it has become very important that websites are creative and innovative if they want to draw in customers. However, websites should not be different just for the sake of being different. Website developers must have a creative side and deliver interesting designs that capture the imagination of users, while still adhering to the principles of good website design. However, there are techniques and certain layouts which can be used to make a website 'stand out' while still conveying information in an appropriate way.

Unconventional layouts

In the past most websites tended to fit to a standard layout template, which, to an extent, is fine because then they

all follow the principles of good website design. However, there still needs to be room for creativity and innovation. A vast number of websites are being created so new websites need to stand out. The art of website design is still evolving and good, new ideas can still be found. Recently, there has been an increase in unconventional website layouts through 'out-of-the-box' thinking on the part of the website developers who came up with them.

Here are some examples of creative and innovative websites.

Ice & Sky

Go to education.iceandsky.com. The Ice & Sky website follows in the footsteps of Luc Jacquet and the Wild-Touch project. What makes this website creative and innovative is that it uses unique animations to help tell the story, combined with an out-of-the-box-thinking layout.

Link

To find out about Luc Jacquet and the Wild-Touch and Ice & Sky projects put these terms into a search engine.

p2 Media

Now go to the p2 Media website at www.p2media.de/. This website makes good use of space by having scrolling pages for the different links within the website. In essence, it uses several home pages and scrolls through each one on a timer. This is an interesting way to present a website and is quite unconventional compared with most websites that you may come across.

Discussion

The internet and paper-based documents are both good sources of information. What other sources can you identify? How reliable are these sources?

Golden ratio

It is not known who discovered the golden ratio, but it is a mathematical concept that has been used for over 2000 years. This ratio has, either intentionally or unintentionally, been used to make designs **aesthetically pleasing**. The golden ratio has been used in:

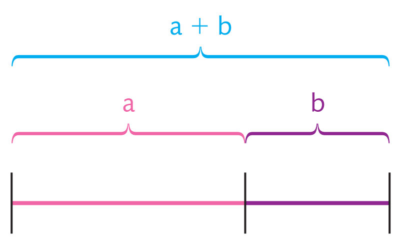
- nature
- science, including astronomy
- art, including painting
- architecture.

The golden ratio can also be applied to website design. This ratio is based on a simple mathematical equation which produces a **ratio**. It produces a special number which is equal to 1.618 (or 1.62 rounded up to two significant figures). This special number is found by dividing a line into two sections so that the longer section divided by the smaller section is also equal to the whole length divided by the longer part: a/b = (a+b)/a or for example 31/24 = (31+24)/31. This number of 1.62 (2sf) can be applied to web design to set the width of the main content and side bar columns.

Key terms

Aesthetically pleasing – something which is visually engaging or appealing to the senses of sight and hearing. Usually such things are considered beautiful or attractive.

Ratio – a way of concisely showing the relationship between two quantities. A ratio is represented by separating the two quantities with a colon (:). For example, a ratio of 1:2 is one quantity compared with something which is twice the first amount (twice as much).



$$\frac{a+b}{a} = \frac{a}{b} = 1.618... = \text{the golden ratio}$$

Figure 15.2: The golden ratio.

Two quantities are in the golden ratio (see Figure 15.2) if the *ratio* of the values is the same as the ratio of their *sum* to the larger of the two quantities. It is often symbolised using phi (Φ), after the 21st letter of the Greek alphabet.

Research

Find examples of where the golden ratio of 1.62 appears within science (including astronomy), art (including painting), architecture and nature.

Search engine optimisation

Search engine optimisation (SEO) is a set of techniques used to maximise the number of visitors to a website. By applying the techniques of SEO, a website developer can ensure that their website appears high on the list of results returned by a search engine for related search terms. SEO is very important to businesses as users of search engines tend to click on the first relevant link they come to. By appearing at the top of a search engine, they are likely to maximise the potential of advertising their products and services. There are books and courses available that can help you optimise your website so that it appears first on search engine results lists, although achieving this is not always as easy as it may at first appear. Outlined here are a number of the techniques that website developers use to optimise their websites for search engines. Remember that Google down-grades non-mobile-friendly sites.

Indexing (meta tags)

Search engines use the entered search term/phrase to find websites in its database which include that term/phrase, rather than searching the whole internet each time. Search engines utilise 'spiders' to trawl the internet for new websites to include in their databases, with the intention that all websites will be included in the databases. The spiders examine each web page that they encounter and send information about that page back to the search engine to be stored in its database. This process is called 'web crawling'. There can be disadvantages to web crawling, because sometimes a web crawler will attempt to spider your website aggressively, which will result in a server overload. Therefore only limited crawling should be carried out.

To ensure that the spiders list a website correctly, website developers include meta tags in the coding of a web page. Meta tags are embedded into the **<head>** section of a web page. These meta tags are used to provide search engines with information about a website. The text in these tags is not displayed within the web page. Instead, it tells search engines (and other web services) specific information about the page. Although the meta tags are not displayed on your website, it is sensible for them to be the same as your web page headings (assuming that these are sensibly titled) to tell people what the purpose of your website is and what products or services you offer.

Keywords

A keyword is a word or phrase that can be put into a search engine so that it will return matching and relevant results. Businesses often use market research companies to see what keywords or phrases best reflect their business, products and services. You should bear the following goals in mind when selecting keywords:

- use keywords that accurately describe the business, products or services
- use words/phrases that people actually type into search engines when looking for something online
- use the most relevant keywords in the URLs and web page titles and to describe the site; for example, you should use the keywords in the meta tags, the titles and body of the text on the pages and even in the image filenames.

Key term

<head> – an HTML tag which is used to provide data about an HTML document (web page).

All this, when used correctly, can help to improve your ranking on the search results page.

Importance of updates

In order to ensure that your website is relevant, it must be updated regularly. Search engines can determine how often your website is updated. If a website is updated regularly, then the search engine considers it a consistent source of new information. As a result, the website is more likely to be placed higher in the search results.

Factors affecting website performance

There are a number of different factors that affect the performance of a website. If performance is adversely affected, websites can load slowly or fail to load completely. If your website does not respond quickly, you are likely to lose users to more responsive websites. Performance is key to the success of a website.

Where scripting runs

Scripting can either run on the web server (server-side scripts) or on the local client machine (client-side scripts). Server-side scripting is used for advanced interactive features such as connecting to a database. It works by the user requesting a web page from the server. The script in the page is interpreted by the server to suit the needs of the user, and then is sent to the device. The downside of server-side scripting is that, when a user makes requests over the network to the server, it can slow down the experience for the user and place more load (strain) on the server. Client-side scripting is when the script is executed on the user's computer, and does not connect to a server. Client-side scripting is useful as it can provide extra interactivity within web pages without the need to connect to a web server.

Browser compliance

All web browsers were not created equally. Each web browser will render code differently. This means that when the code of your website is loaded within a web browser it can be interpreted in a different way by that web browser to the way it would be interpreted by another web browser. For example, how your website looks in Firefox® may not be how it looks in Internet Explorer®. Some pages will load faster than others depending on which web browser you use. In addition, some elements of web pages may not be supported, depending on which web browser you use. As a website developer you may be asked to develop a website which is compliant for two or three web browsers. Your job would therefore be to develop a website that appears consistently in all of these web browsers. Different web browsers all have their own advantages and disadvantages. For example, Chrome™ is good for HTML5 support, whereas Firefox® is good for its website developer add-on options. It is important to realise that, when using a web browser, there is not necessarily an even playing field. Each one is different and unique in its own way.

Server-side factors

Some of the factors affecting the performance of websites are server-side: that is, they relate to the capabilities and capacity of the web server being used and whether it has been bought or is rented.

- Bandwidth availability – This determines how much traffic can be handled by the web server; specifically, how much content can be downloaded at any one time. Bandwidth can be thought of as a pipe from the web server to the users. The bigger the pipe, the more that can be sent down it. Conversely, the larger the content to go down the pipe, the fewer bits of content that can go down it at any one time. The larger the web page and its associated files, the less users can download from it at any one time.
- File types – By using smaller file types which use **compression** methods, the website will have a faster download time. When deciding on which file types to use, a website developer must make a judgement in order to balance quality and file size, because the higher the quality, the larger the file size.
- Number of hits – The number of web page hits can have an effect on web page performance. For example, if too many people are on the same web page at the same time, then it can overload the web server and slow down the website's performance.

Key term

Compression – where a mathematical calculation is performed on a file in order to 'squash' it and make it smaller.

Client-side factors

Some of the factors affecting the performance of websites are client-side: that is, they relate to the capabilities of the user's computer system. If the capabilities of the user's system are poor, those people using that system might not be able to access certain websites, particularly those that suffer from poor server-side performance.

- Upload and download speeds – The speed of the user's internet connection will determine how quickly they are able to download or upload web pages.
- Processor speed – As the connection speed will determine the rate of download, so the computer's components will affect the speed with which the content is displayed and with which users can interact with it. You must take into consideration that a user's device may not have a fast processor or large memory capacity and so website developers must decide between high user specification requirements and a high number of visitors. Sites that rely heavily on client-side scripting such as JavaScript can put a significant load on a user's CPU.

Table 15.2: Connection types with descriptions

Categories	Connection Method	Description	Typical Speed
Fixed line narrowband	Dial-up	This was the traditional method of connection. It uses the existing analogue telephone lines and it remained popular for many years. However, in 2013 BT turned off dial-up internet access service, in favour of broadband.	56 Kbps
	ISDN (integrated services digital network)	ISDN was used to generate faster speeds than dial-up. It still used a phone line to which digital lines needed to be connected. However, ISDN is slowly being phased out for much faster connection methods.	128 Kbps
Fixed line broadband	DSL (digital subscriber line)	Using digital lines, DSL was introduced. It is the most common method to provide a broadband service. The most common in the UK is ADSL (asynchronous digital subscriber line).	1 Mbps to 8 Mbps
	Broadband	This technology is constantly being developed and faster speeds are conceivable in the near future.	Cases of up to 100Mbps in England
	Fibre optic	This connection method utilises visible light as a transmission method. This is significantly more efficient and reliable than typical broadband which uses copper wires to send electrical signals.	Up to 1000Mbps
Wireless broadband	Mobile broadband	Mobile broadband is a wireless communication technology that uses mobile phone networks and can generate internet access to almost anywhere. This service is usually accessed through a mobile phone (smartphone).	Internet speeds vary. 384kbps to 30Mbps for 3G (3rd Generation). 4G speeds can go to 1000Mbps (1Gbps)
	Wireless hotspots	Wireless hotspots are places such as shops and cafés, which offer you free access to their broadband connection. You may need to be a member to get the password for the wireless connection.	Depends on the internet service provider and how many people are logged on

Research

Working in small groups or in pairs, research the differences between fibre to the home (FTTH) and fibre to the building (FTTB). How do these technologies work? What are the advantages and disadvantages of each?

- Cache memory – Cache memory may be the memory that your computer has. The bigger the memory the faster the computer will run. The cache memory is used to reduce the average time needed to access the memory. This means that the more memory you have, the quicker your computer is at dealing with the website you are accessing. The cache used by websites may be on the physical storage (e.g. hard drive in the form of temporary files) as well as within the computer's RAM.
- Browsers – As discussed in the section on Browser compliance, web browsers were not created equally. Some web browsers can affect the speed at which a web page can load.
- Interactivity – When you incorporate interactive elements into a web page, such as Flash images or games which are embedded, this can affect the speed at which the website can be loaded. The more interactivity a website has, the more time is needed for your computer system to download all the information. With new technologies such as 4G, 5G and fibre optics, this is less of a problem. Previously, it took time for the information to download and, consequently, was used sparingly. Now interactivity is standard and users tend to expect it within websites.

Assessment practice 15.1 A.P1 A.M1 A.D1

You have applied for an apprenticeship with a website development company, where you hope to prove to the recruitment manager that you are a competent website developer. Competition for the job is high, so you have been set a preliminary report-writing task to gauge your understanding of website development principles. Applicants who perform well in this task will progress to the next stage of the recruitment process.

You are to identify two websites with a similar theme, and compare the principles of website design that each website employs. It has been recommended that you take screenshots of any part of the website if you wish to back up your reasoning. You must conclude your report by analysing each website's suitability for their intended audience and purpose.

You should take this further and fully evaluate the way in which the principles of website design are used to produce creative, high performance websites which meet client requirements. You are free to choose which websites to use within your evaluation.

Plan

- What is the task? What am I being asked to do?
- How confident do I feel in my own abilities to complete this task? Are there any areas I think I may struggle with?

Do

- Have I spent some time planning my approach to the task at hand?
- Am I confident that I know what I am doing and that I know what it is I should be achieving?

Review

- I can explain what the task was and how I approached it.
- I can explain how I would approach the hard elements differently next time (i.e. what I would do differently).

B Design a website to meet client requirements

Once you understand the principles behind good website design, you will need to put this knowledge into practice by designing a website.

Website design

The website design process involves a number of steps which, if they were to be skipped, would cause major problems when the website **goes live**. There is a saying: 'If you fail to plan, you are planning to fail.' This saying very much applies to website development, but also to any design and development project.

Key term

Goes live – describes the first time a website has been uploaded to a web server and is made available to the public.

Problem definition statement requirements

Before a website can be built, a website developer must elicit as much information about the requirements of the website as they can from the client. The more information a website developer has about the client's requirements, the more likely they will be to fulfil the requirements effectively. All the information about the requirements for the website will be collated within a problem definition statement.

Intended audience

Web designers must always have two sets of needs in mind: those of the client and those of the users. The client is the person who has commissioned the website and, usually, they are also the person who holds the purse strings. If the client is not happy with the website, you (the website developer) may not get paid for your work.

The users are the visitors to the website. They need to be attracted to the website initially, and then encouraged to revisit to make more purchases, to look at new content or take part in discussions on forums. One aim of websites is to persuade their users to bookmark the website, which increases the probability of them returning on a regular basis.

Full summary of the problem to be solved

A website developer will need to understand the full problem to be solved. This will require communication between the developer and the client. There are a

number of ways in which this can be done. A commonly used method of requirements gathering is called SQIRO. SQIRO stands for sampling, questionnaires, interviews, research and observations. A website developer will use these techniques with the client so as to gather as much information as possible to fully understand the problem that the creation of the website is intended to solve.

Constraints

A client will need to understand the constraints of the website. The constraints are the limitations the website could have. Typical constraints of websites often include:

- money
- timescales
- staff training
- levels of security
- support and maintenance contracts.

Tip

Within the IT industry, you are unlikely to ever be working alone. Most websites in the industry are built by teams, with different people working on different elements of the design and development. For instance, you may have a graphic designer working on images, a software tester, a website coder, a designer and so on. With so many people working on the same project, it is crucial that people learn to work as a team and support each other. If you do not work well together in a team and do not communicate well with each other, it is likely that the resulting website will not meet the client's requirements or will not be delivered on time.

Benefits

The reason why a client will approach a website developer is for the potential benefits that a website will bring to their business. The benefits for a business of having a website include:

- advertising for your products/service and of the business worldwide
- less expensive than printed media, radio and television advertising
- more accessible (24/7, 365 days a year) (you do not need to turn away customers because it is closing time since a website is accessible to customers at all times)
- other websites may link to you, spreading the word about your business's products or services
- it gives you the opportunity to gain long-term clients. (There is a difference between a client and customer. A customer is someone who walks in and buys something. A client is a regular customer who will often return to your website, which can help your business grow.)

Nature of interactivity

Most modern websites involve interactivity: that is, they are **interactive websites**. You risk losing users if you have only a **static website**. It is important to decide how much interactivity will feature on a website. Too little interactivity and users may lose interest, too much interactivity and they may feel overwhelmed. It is important to get the balance right.

For e-commerce websites, the web designer also needs to decide how online transactions will be performed. There are two parts to this issue.

1. How will the user browse the catalogue? For example, how will the items be listed in the catalogue? Consider e-commerce websites such as Amazon™ (Amazon and the Amazon logo are trademarks of Amazon.com, Inc. or its affiliates). Amazon™ has items which are catalogued in a hierarchical fashion, where you can filter your preferences, such as the relevancy, date added, price (in ascending or descending order) and customer reviews.
2. How will users make purchases? Before a user will purchase an item, the website will need to register their details and credit/debit card information. This information is encrypted using a method called **transport layer security (TLS)**. PayPal™ uses such a method because it stores all your financial information, such as credit card details.

Research

Research the terms server-side and client-side scripting. What is the difference between the two techniques? In pairs, see if you can draw a diagram that represents how these two techniques work. Look back at Factors affecting website performance to get you started.

Key terms

Interactive website – involves some level of activity from a simple feedback form to a database that personalises the website for each individual visitor. Changes can be made to the website 'on-the-fly'.

Static website – one with no interactivity, which is usually just a presentation of information. Changes to the website have to be hard-coded into the website.

Transport layer security (TLS) – is a protocol that makes certain that there is privacy between communicating applications and their users on the internet. TLS is the successor to the secure sockets layer (SSL) method.

Purpose requirements

Once all the available information has been gathered, the requirements are then generated. It is important that all the requirements that are gathered are SMART. When we refer to SMART requirements, we are saying that the requirements are as follows.

- **S**pecific – Targets a specific area of improvement.
- **M**easurable – It is possible to measure whether a requirement has been completed, that is, you can verify its completion. It is best to avoid any requirements that cannot be verified as complete.
- **A**chievable – The requirement must be achievable.
- **R**ealistic – The requirement can be realistically achieved with the available resources.
- **T**ime-constrained – The requirement can be achieved within the timeframe allocated.

Design ideas and prototyping

By this stage, you should understand the client's requirements for the website and what it needs to do. Therefore you can use design tools to design a website which can be presented to the client for approval. Once the design has been approved, it can be used to create a website that goes live. It is important to consider the principles of good website design when you create a website and ensure that these are incorporated within it.

Link

Look back at the Principles of website design to remind yourself of the fundamental principles which should be applied to any website design.

When developing any IT product, including websites, it is not uncommon for developers to build a prototype. This is usually done so that the user can have an idea of what the website will look like as well as what it will do. The idea for prototyping comes from engineering, where a prototype is often built before the construction on an assembly line to produce many copies.

Several tools can be used to ensure that all areas are considered when designing websites. By producing a thorough design, using the tools presented in this section, and using this design to communicate with your client, you can ensure that your client is happy with your plans before you build the website. This should reduce the problems that you would encounter if there was a mismatch between client expectations and the actual outcomes.

Diagrammatic illustrations

To convey the concept of a website, you might use one or more of the following tools.

Mood boards

The aim of a mood board is to produce something with the same feel as the website (see Figure 15.3). It is a useful way of focusing the design and demonstrating initial ideas to the client. Think of a mood board as a sketchbook where you can collate images, different typographies, fonts, artwork, sample layouts and so on. The mood board can be presented to the client who can decide what they like and do not like on it. Their feedback and amendments to the mood board can then help to focus the website developer on what the website should look and feel like.

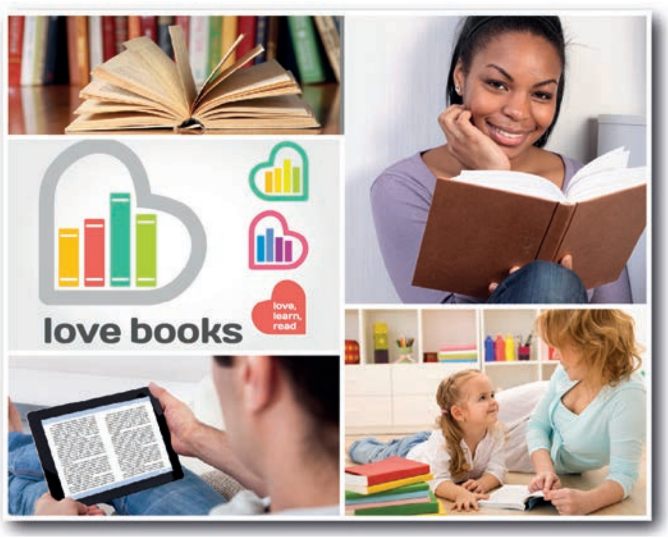


Figure 15.3: Mood board picture example

PAUSE POINT

Think of a website that you would like to develop. It could be for your favourite TV programme, your favourite computer game or an idea that you have for a business. For this website develop a list of:

- 10 SMART requirements for the website
- 10 website requirements which are not SMART.

Hint Working with a peer, compare your requirements and work together to see if they are correct or not.

Extend What do you feel would be the implications for the design and development of a website if you did not have SMART requirements governing it?

Storyboarding

Storyboarding is key to structuring a website clearly and is a way of expressing a navigational design. Storyboarding is not just used in website design; it is often used in the design of moving images such as animations or films.

Realistic representation

Your initial designs should give a realistic representation of what the website will look like. This is where it is important to be in communication with your client so that, upon seeing all your design documentation, the client can envisage what the website will actually look like.

Wireframe

To visualise what the pages will look like before building them, designers create screen designs or wireframes (see Figure 15.4). Wireframes are mock-ups of the actual pages, concentrating on layout rather than content. They also usually include some of basic **attributes** of the pages.

Key term

Attributes – additional information about your wireframe. For example, which font you will use, which font size you will use, where the sidebar will appear, alignment and so on.

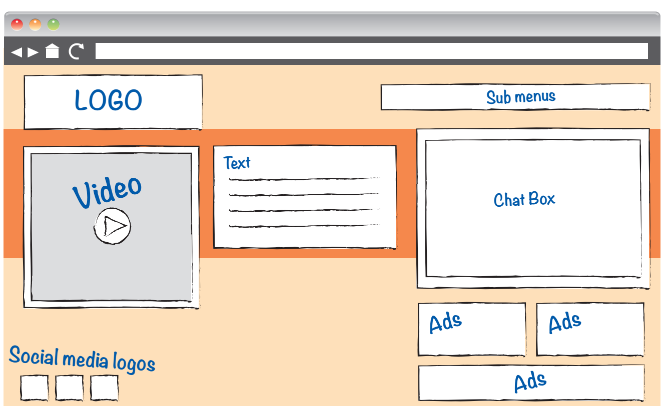


Figure 15.4: Wireframe artwork

Site map

A site map (see Figure 15.5) is a list of pages of a website that is accessible to crawlers or users. It can be a document, in any form, which is also used as a planning tool for the website design, or a web page that lists the pages on a website. It also shows the relationship between the pages, representing how they are linked together.

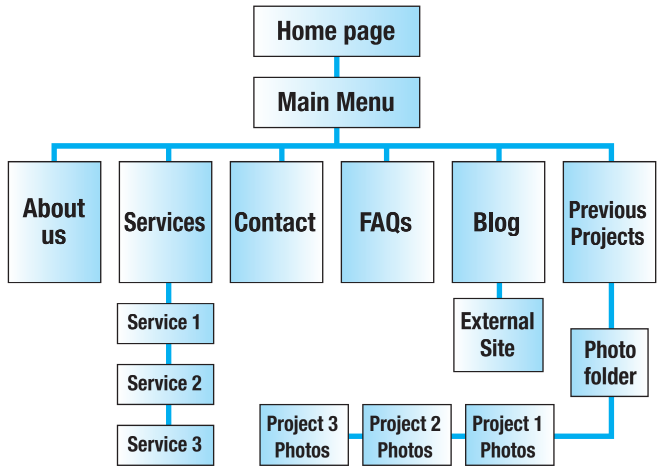


Figure 15.5: A site map

Search engine optimisation

When designing a website, the developer will need to consider the best way to maximise the number of users who visit the website. Therefore, it is important that the design conveys an effective solution which users will want to keep returning to. Websites which receive more traffic are more likely to be ranked highly in search engine results pages, which will be beneficial to the business because users are likely to click on the first relevant link they see. A good design is not in itself enough to get more users visiting a website, although it might mean that those who find the website will return. To get more users visiting your website in the first place, you need to employ SEO techniques to ensure that your website appears at the top of search engine results pages.

Link

For more on SEO see Search engine optimisation.

PAUSE POINT Think of a scene from your favourite film or cartoon. Create a storyboard to show what happens. Use a minimum of 10 boxes and a maximum of 20. Add any notes underneath each box to explain what is happening.

Hint Most storyboards tend to be sketched out on paper. You can use Microsoft® Office® packages such as Word® and PowerPoint® to develop storyboards using the Insert feature ribbon.

Extend Add colour to your storyboard to show contrast and to give your storyboard a more realistic representation.

Alternative design ideas

Alternative design ideas should be considered and thought through in case problems are identified with your preferred design and also to give clients more choice in terms of the design they decide to go with. Alternative design ideas can be presented within your mood board and in storyboards. Once the client is happy with a particular design, you can focus on this design in a wireframe. Consideration must be given to design compatibility with mobile devices; this is something that will need to be discussed in the requirements stage. It is common these days for businesses to develop mobile-friendly versions of their websites. However, for smaller businesses, it might not be possible to develop a separate mobile website so the one website needs to be suitable for viewing on desktop computers and on mobile devices. With HTML5 it is now more common to develop a responsive website that dynamically adjusts elements to work in any size browser/screen without having multiple versions to maintain.

Discussion

Using your PC, go to **uk.pearson.com** and notice the design (see Figure 15.6). Now using your smartphone or tablet device, go to **uk.pearson.com** (see Figure 15.7). Can you see the difference? Notice how the two websites look different despite being the same website. This is done for user convenience and compatibility. Discuss, in your class, how you think a website developer would go about making this possible.

Client-side scripting design

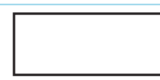
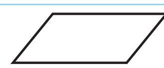
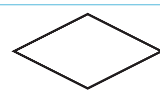
Client-side scripting refers to websites where the script is executed client side (by the user's web browser) instead of server side (on the web server). JavaScript is a programming language that enables client-side scripting and can be used to create interactivity within web pages.

Prior to coding or scripting, appropriate designing must take place. This must be done to inform a website developer of the function that the website must perform. When they have all available design documentation, the website developer can then create the website.

Flowcharts

Flowcharts are diagrams that are used by software developers to represent a solution to a given problem. Flowcharts involve boxes of different kinds, used to represent different things, and arrows that show how the boxes relate to one another (see Table 15.3 and Figure 15.8).

Table 15.3: British Computer Society (BCS) flowchart symbols with descriptions

Flowchart symbol	Description
	An oval shape represents a start or end point
	An arrow is a connector that shows the relationship between representative shapes
	A rectangle shape represents a process
	A parallelogram represents where an input or output will take place
	A diamond shape represents where a decision will be made

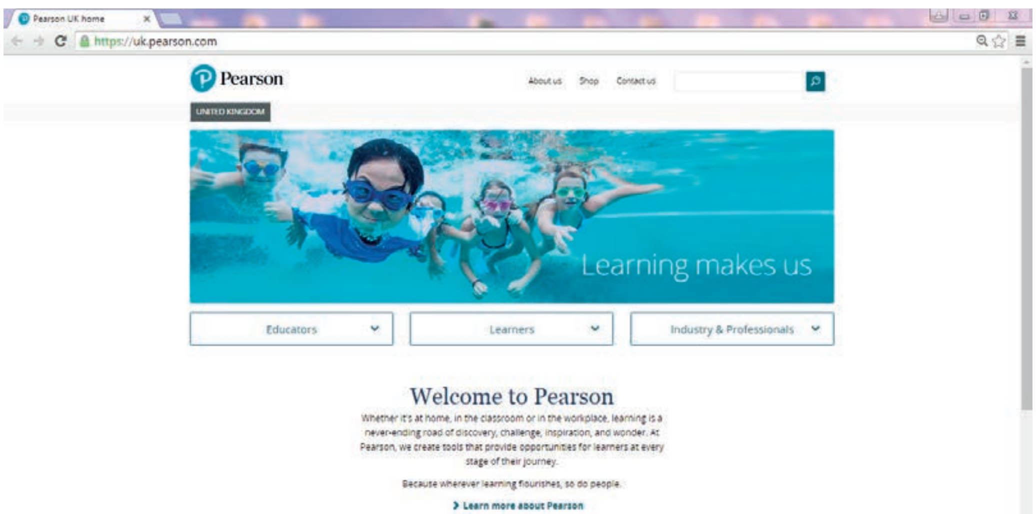


Figure 15.6: uk.pearson.com (desktop view)

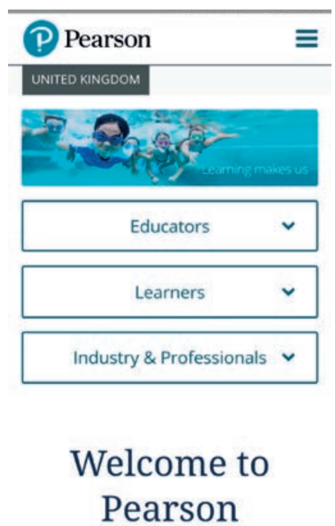


Figure 15.7: uk.pearson.com (mobile view)

Tip

Decisions are essentially IF statements. Therefore, there will always be a minimum of two arrows coming from a decision (diamond), each one representing a decision, e.g. Yes or No.

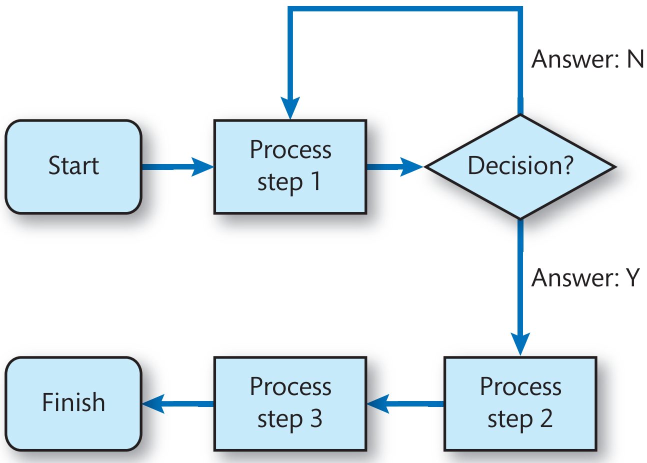


Figure 15.8: A flowchart example

Pseudocode

Pseudocode helps programmers develop **algorithms** by forming an intermediary step between an explanation in English (or another natural language) and the coding language. By using pseudocode, a designer can plan what the code will do, without having to worry about ensuring that the correct words and syntax are used.

Key term

Algorithm – is a procedure or formula for solving a problem.

Ready-made and/or original assets

Assets include animations, graphics, audio and videos. All of these assets can be embedded within a web page to provide additional interactivity, information or context.

Ready-made assets

It is important, if you are using pre-existing, ready-made assets, that you seek the owner's permission to use their assets on your website, as there could be copyright issues. If you do not seek the owner's permission to use an asset, you will be in breach of the UK's Copyright, Designs and Patents Act 1988. The maximum penalty for breaking this law is a possible prison sentence and/or a large fine. Therefore it is essential that you get permission from the owner of any asset that is not copyright free before using it in your website. Getting permission will sometimes, but not always, involve paying a fee.

Original assets

It is simpler to use original assets: that is, ones that you have created yourself from scratch. Any animations, graphics, audio or video that you create yourself can be embedded within your website without any possibility of infringement of copyright, as long as they are not based in any way on someone else's work. (You are the copyright owner of these assets.) You do, however, need to be careful of certain issues when creating your own original assets, for example photographs. If you take photographs of people, you need them to sign a model release form that gives you permission to use the photograph that they appear in as you wish.

Obtaining and using feedback from others

When you have completed your website designs it is important to gather feedback from others (including the client and potential users) to see if the designs meet the requirements and could be refined in any way to make them better. This is a crucial part of the design and development process because if issues were found with the design after the developer starts building the website this is likely to lead to delays. Also the website may not be as good as it might have been, because time and cost constraints might mean that compromises have to be made. Therefore, it is important to obtain feedback on designs before they are implemented, to determine if any refinements need to be made.

The feedback stage is also an opportunity to identify any technical and design constraints. By identifying them at this stage, you will be able to come up with alternative

Table 15.4: Pseudocode example and JavaScript equivalent

Pseudocode	JavaScript
Age = input from user	`Age=prompt("Enter age", "");`
If age>= 18 then	`If (age>=18)`
Print onscreen "Hello World"	`{document.write("Hello World");}`
Else	`Else`
Print onscreen "I am x years old"	`{document.write("I am " + age + " years old");}`

design solutions to overcome any constraints, before it is too late to do anything about them. In addition, the website development process is recursive, meaning that you can go back and adapt the designs to factor in the feedback of others before going ahead with development, not just once but repeatedly until the design is right (keeping in mind time and cost constraints, of course).

One method of obtaining feedback is to use a questionnaire. Questionnaires can be used to gather feedback and determine areas of strength and areas in need of improvement. Questionnaires use two main types of questioning, **quantitative** and **qualitative**.

Examples of quantitative questioning

- On a scale of 1–10 how would you rate the overall aesthetics of the website?
- Could you navigate through the website effectively? (With answer options such as 'Yes, easily', 'Yes', 'Yes with difficulty', 'No', 'No, it was very difficult'.)
- Did you find the pop-up boxes a distraction or were they beneficial? (With answers options such as 'Distraction' or 'Beneficial'.)

Examples of qualitative questioning

- What did you enjoy most about the website?
- How could we improve your website experience?
- Do you have any feedback about the website you think we should be aware of?

Key terms

Quantitative questioning – questions have a definitive answer, either a numerical value or specific answers in ranges. It focuses on statistical analysis.

Qualitative questioning – questions do not have definitive answers. It provides answers as to how or why and is used to gauge opinions and get more detailed feedback.

Link

For more on quantitative and qualitative questioning see Test users and user feedback in *Unit 17: Mobile Apps Development*.

Using the test information

It is recommended that you get several people to test your website. This is because someone may identify an issue not noticed by yourself or your other testers. However, you cannot test forever as you will be bound by time and budget constraints. Therefore, once you have the information, you must use it to refine the website, where necessary. Any errors that are spotted must be changed. For example, if a link within the website does not work as expected, this must be resolved. It is also an opportunity for the website to be tested on different web browsers

PAUSE POINT

Part 1:
Create a wireframe for a website based on a theme of your choosing. This wireframe will be for the homepage. Be sure to include all necessary attributes in your design and to apply the principles of website design.

Part 2:
Develop a questionnaire for potential users about your homepage which poses five quantitative questions and five qualitative questions.

Part 3:
Ask a peer in your group to complete your questionnaire about your homepage. They should answer your quantitative and qualitative questions and critique your wireframe design, citing any areas in need of improvement. (If you are able, ask more than one person to complete your questionnaire; this could include family and friends.)

Part 4:
Based on the questionnaire feedback, make any necessary or beneficial refinements to your wireframe design for the homepage.

Hint If you are developing your wireframe using a package such as Microsoft® Word® or Excel®, remember to change the page orientation to landscape to give you more room. Also, annotate your designs effectively using the Comments feature within the Review section of the ribbon.

Extend Explain how your website fulfils the fundamental principles of website design. Annotate your wireframe design indicating where you have adopted these principles.

and on different mobile devices. Different web browsers render code in different ways, which is why some websites look different depending on which web browser you are using.

This test data is invaluable, as you can then refine the website and streamline it to make sure that it is fit for purpose and meets the needs of the client.

Discussion

In small groups or in pairs, discuss why it is important to get other people to test your website. What do you feel are the implications of not doing so? Furthermore, what are the potential problems of testing a website yourself only?

Testing plan

A testing plan is used to test functionality. In other words, a testing plan is used to check that all parts of the website work as they should. Sometimes, this is referred to as 'black box' testing as it focuses on functionality as opposed to the internal mechanics or workings of a program. This is where, prior to the website being built, a series of tests or test data is developed so that, when the website *is* built, the test data can be used to see if all parts of the website work as they should.

Research

Compare black box testing to white box testing. What are the differences between the two?

Link

For more on black box testing see Testing a mobile app in *Unit 17: Mobile Apps Developments.*

The elements that occur within a test plan are:

- test number
- purpose of test
- additional data on test, eg which web page
- test data
- expected results.

Tip

When developing your test plan, change your page orientation to landscape. This will give you more space and enable you to type more.

Example testing plan

Remember to test all parts of the website and not just those elements that appear on the home page. It is important to test a range of features and attributes of the web pages, not just the links. For example, you should test whether images appear but also test for browser compliance: that is, ensure that it looks correct in different browsers. See Table 15.5 for part of an example test plan.

Table 15.5: Example test plan

Test number	Purpose of test	On page	Test data	Expected result
1	Test the home button	about.html	Left click	Load home.html
2	Logo.gif	home.html	Load page	Appears in the centre of the page
3	Test the alt/title tag of Logo.gif	home.html	Hover over Logo	Tool text tip appears

Link

Refer to *Unit 4: Software Design and Development Project* for more details on testing.

Technical and design constraints

Constraints are limitations or restrictions that may make it more difficult to design and develop a website. They may limit the scope or complexity of the website you design. Constraints tend to fit into two categories: technical and design constraints. Some of the possible constraints are outlined here.

PAUSE POINT Develop a series of 10 tests which could be used to test the functionality of a website of your own choosing.

Hint If you are not comfortable using tables in Word® to do this, consider using Excel®, which is just as effective.

Extend Work with a peer within your group to see what additional tests they have thought of, and apply these to your test plan. This will help both of you to have a wider range of tests in your test plans.

Technical constraints

- IT staff – The skills needed for the creation of the website might not be available, meaning that staff would need to be trained or recruited to complete the required design and developent work. Moreover, there may be full time staff who will need to update, and manage the content of the website. Therefore this is another technical constraint which needs to be considered.
- IT equipment – The website might require specialised hardware or software that may need to be bought in or installed before the website design and development can proceed.

Design constraints

- Financial – The client's requirements for their website might not be feasible within their budget. A budget will have been agreed with the client and you have to work within that budget. However, if the client wants features that would cost more than the budget, you can present this information to the client. They may choose to increase the budget so that they can have the additional features or they may compromise on the features to keep the budget the same.

Theory into practice

When undertaking a project which requires you to be time constrained (it needs to be built by a specified date), it is important that timescales are built in to ensure that the final website is completed on time.

When working in the IT industry, it is extremely likely that you will be given tasks that need to be completed within a specified time frame. Therefore, you have to set relevant targets with deadlines for when each target needs to be achieved. This must include contingency planning, such as taking into consideration the feedback from others (and so having to make improvements), as well as the possibility that you might be ill for a few days.

1. Write a list of sensible targets which could be given specified deadlines for the process of designing and developing a website: that is, milestones within the design and development.
2. Once you have this list of milestones, allocate amounts of time for the completion of each of these stages.
3. Put each stage into a schedule, using the amount of time allocated for each stage, to determine the dates by which they would need to be completed (using an end date of your choice).
4. Write a list of factors that might affect your ability to complete your milestones on time, such as being ill.

- Flexibility – You need to design a website to work on different platforms (both on desktop computers and mobile devices), then you will need to consider this constraint when designing the website.

Legal and ethical considerations

We need to consider the legal and ethical considerations involved when designing and developing websites. For example, there are laws that protect an individual's intellectual property and those that ensure that an individual's personal data is not released.

Copyright, Designs and Patents Act 1988

The Copyright, Designs and Patents Act 1988 protects all original works such as music, art, writing and programming code once it is tangible, which means once it is in a fixed form (for example, a music album is released or a book is published). These original works are the intellectual property of the individuals who created them. As the internet has become such an important part of everyday life, the question of whether websites are subject to protection under copyright laws has often been discussed. It is now accepted that a website becomes tangible once it is coded and saved onto storage media, so websites are now protected by copyright.

Data Protection Act 1998

The Data Protection Act 1998 was designed to protect sensitive data held in databases. It was originally passed in 1984, with an update in 1998, which was brought into effect in 2000. It is upheld by the Information Commissioner's Office. Every business that stores data (that is, information about customers), must register and state the data they plan to hold.

There are eight principles of the Data Protection Act 1998 (see Figure 15.9). The data subject is the person to whom the data refers. Under the act, the data subject has several specific rights, including:

- the right to compensation for unauthorised disclosure of data
- the right to get inaccurate data corrected or removed
- the right to access data and apply for verification or erasure where it is inaccurate.

Prior to development, a website designer must consider how the Data Protection Act could affect their website. For example, a website that collects users' data, such as an enquiry form asking for personal information (eg your surname, forename and phone number) should include a Privacy Policy that informs website visitors how you retain, process, disclose and purge their data in line with the Act.

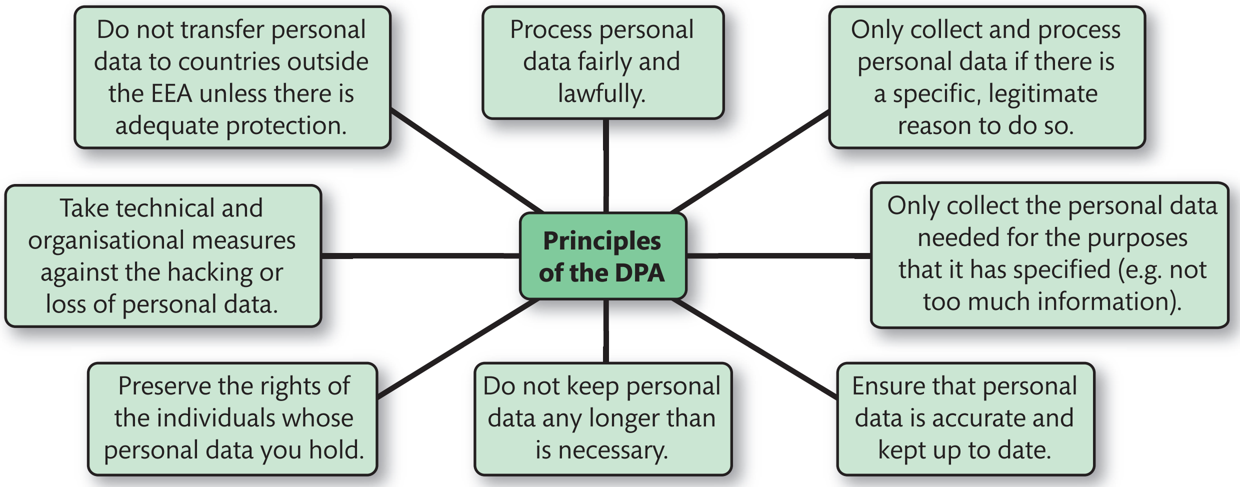


Figure 15.9: The eight principles of the Data Protection Act 1998

PAUSE POINT

Think about a website with which you are familiar, such as your school or college website.

How do the laws listed affect the content and running of the website?

Hint Consider working with a peer in your group to exchange and collaborate on ideas.

Extend Consider what would happen if your school or college website did not abide by these laws. What would be the consequences?

Common tools and techniques used to produce websites

It is important that, once appropriate designs have been approved and completed, different tools and techniques for development are explored in order to meet the client requirements.

HTML

HTML (hypertext markup language) is the most commonly used markup language, so much so that all others are just about extinct. It forms the basis of all worldwide web pages, even if other languages are used for parts of them.

HTML uses a system of tags (indicated by angle brackets < and >) which contain the instructions. Almost all instructions come in a pair of open and closed tags enclosing the content to be affected, for example Some text would produce 'Some text'. Note that American spelling is used in HTML. HTML pages should start with <html> and end with </html> tags to declare the language being used.

HTML5

HTML5 is the current hypertext markup language standard used for structuring and presenting content on the worldwide web. Website developers should always work to the current standard so, when developing your web pages, you should code using this standard. Tags for changing the font are no longer required in HTML5; instead, cascading style sheets (CSS) should be used. It is likely that there will be subsequent updates, leading to HTML6, which will then become the new standard that website developers should use.

HTML5 was finalised by the **World Wide Web Consortium (W3C)**. HTML5 has a larger set of technologies that allows for more diverse interactivity and more powerful websites and applications.

Link

For an introduction to HTML5 elements and tags, see HTML5 Introduction at www.w3schools.com

Key term

World Wide Web Consortium (W3C) – an international community that develops open standards for the use of HTML5 to ensure the long-term growth of the worldwide web.

Tables

Tables used in HTML begin with the `<table>` tag. Tables should not be used to structure websites (which has been done in the past), because it causes accessibility and browser rendering issues. Instead, the `<table>` tag should be used only to present a table on a web page. For example, if you are presenting the opening times of a shop.

Forms

Forms are used in website development to collect user input. There are different ways of collecting user input using forms.

- Text field – This defines a one-line input field for text input.

```
<form>
  Please enter your first name:<br>
  <input type="text" name="firstname">
  <br>
  Please enter your age:<br>
  <input type="text" name="age">
</form>
```

- Text area – Allows you to have more user input.

```
<form>
 <textarea name="textarea"> Please enter your text here</textarea>
</form>
```

- Submit button – Defines a button for submitting a form to a form handler.

```
<form action="test_page.php">
  Where were you born?:<br>
  <input type="text" name="birthplace">
  <br>
  Where do you currently live?:<br>
  <input type="text" name="livingplace">
  <input type="submit" value="submit">
</form>
```

- Radio buttons – Radio buttons let a user select one of a limited number of choices.

```
<form>
  Are you sure?
  <input type="radio" name="validation" value="yes" checked>YES
  <br>
  <input type="radio" name="validation" value="no">NO
</form>
```

- Check boxes – As opposed to radio buttons, which only let you select one of a limited number of choices, check boxes allow you to select more than one choice.

```
<form>
  I own the following:<br>
 <input type="checkbox" name="games">Playstation<br>
  <input type="checkbox" name="games">XBox<br>
</form>
```

Navigation

Navigation is a way of moving around a web page to find what you need. At first, the worldwide web was restricted to simply using hyperlinks. However, as time has progressed and technology has advanced, new ways of navigating a website have been developed.

- Menus – Menus can appear anywhere within a web page, but most website developers tend to place them towards the top of a web page. Instead of traditional hyperlinks, these menus are more attractive and aesthetically pleasing. Visitors to a website can click on a menu and it will redirect them to the page they are looking for. Alternatively, sometimes when you hover over a menu a sub-menu will appear (see Figure 15.10).

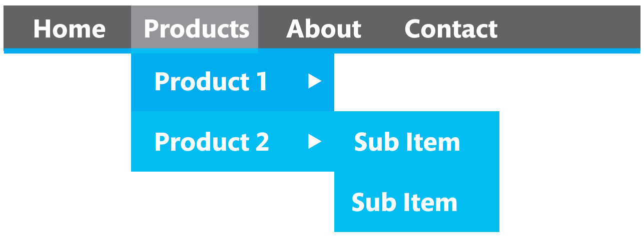


Figure 15.10: Menu and sub-menu

- Hyperlinks – Hyperlinks are links that, when clicked on, take you to a particular part of the website (an internal link) or they will take you to another website (an external link). Hyperlinks can be text, images or buttons.
- Anchors – Anchors are used to redirect a visitor to a certain part of a web page. For example, if you are reading a long document online, at the beginning of the web page there will be links to the beginning of each section within the document. By clicking on one of these section links you will be redirected to the precise point in the web page that you require.

Interactive components

Interactive components are used as a means of enhancing the look and feel of a website. However, a web page should not be littered with too many interactive components because they increase the download time and can make a website look amateurish. Interactivity involves two-way communication between the user and the computer. In other words, it requires input from the user which provokes a response from the computer. This could include giving feedback, searching a catalogue of products or purchasing a product from a website. To have a full catalogue of products would require a database and server-side scripting.

Research

What are the components involved in server-side scripting? Research, compare and contrast 'PHP' with 'ASP'. Which do you believe is best suited for server-side processing?

- Hot spots – A hot spot is an area of an image that acts as a hyperlink. When a person clicks on a hot spot, a hyperlink takes the user to another web page. In addition, hot spots can have rollover capabilities..
- Pop-ups – These are small internet windows that pop up on your screen to get a user's attention (see Figure 15.11). Sometimes they can be considered annoying or dangerous. They are often used by advertisers. The first

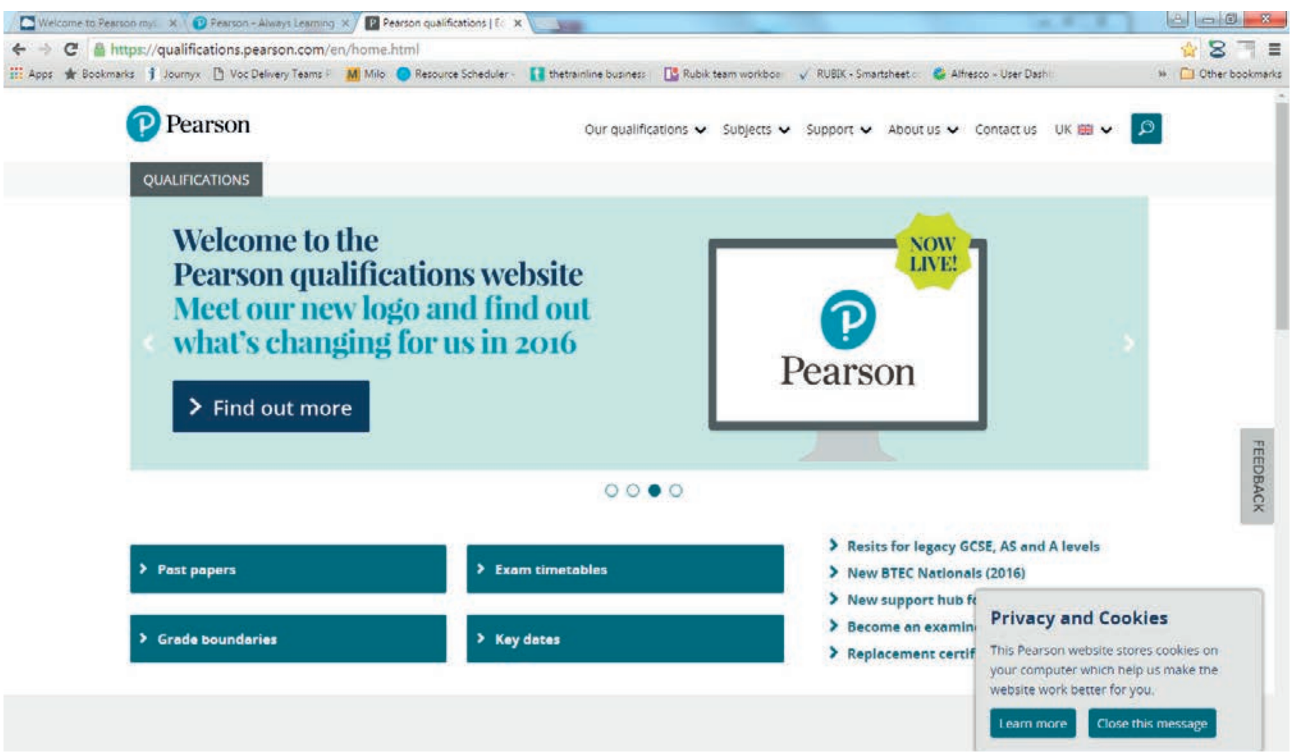


Figure 15.11: Cookie pop-up on Pearson PLC Qualifications website

time you visit a website, a pop-up will appear to tell you that cookies for that website will be downloaded to your PC (this is now a legal requirement) and you have to accept them to continue using the website. The cookie that is downloaded remembers that you have visited the website, so the next time you visit it the pop-up will not appear. However, the cookie might have tailored the website based on any customisable choices you made last time you visited.

- Buttons – These look like command buttons in that, when you press them, they will depress like a conventional button. Buttons have different purposes, one of which can be to take you to another web page.
- Rollover images – A rollover image is a secondary image loaded into your web page to display when a user on your website rolls their mouse over (rolls over) a certain image within your website. It is used to make your website more interactive. For example, some clothing websites display one image of a piece of clothing but when you roll over the image another view of the piece of clothing is shown.

Colour schemes, styles and templates

- Modern web design, within HTML5, uses templates with built-in colour schemes and styles. There are templates which are free to download and some that you have to pay for. These templates are built using a technique called cascading style sheets (CSS), which is discussed further in the next section.
- What makes an appropriate or pleasing template is subjective, that is, a matter of opinion. Quite often the colour scheme and styles used by your website will determine its theme. Therefore you must choose appropriate colours and styles for the type and content of your website. For example, if you are producing a website for a nursery, it would make sense to use a playful font and a variety of bright colours, whereas, if you are producing a professional corporate website, then a more minimalistic look using a small colour palette and contrasting backgrounds is considered best.

Discussion

Investigate different colour combinations that you could use for a website which is education based. Think about accessibility issues such as colour blindness. Discuss, in a small groups, which colour combinations would be best suited and explain your reasoning. Use the BBC website for more information.

Cascading style sheets

It is good practice to use the same layout and styling throughout a website. Cascading style sheets (CSS) are used to ensure standardised formatting across a website. CSS allows you to create a standard layout and style which can easily be applied across all the web pages within a website. They are cascading in that when you make a change in one place within a website this change will be cascaded to all the web pages within the website that use that style. This makes altering and maintaining the design of the website much easier. For example, in HTML to change the font colour of all the titles in a website to red would have involved changing each one individually. But now, using HTML5 and CSS, only one value would need to be changed, and the change would be immediately applied throughout the website for every title.

Link

For more on how to apply CSS to web pages see Using cascading style sheets.

The World Wide Web Consortium

The World Wide Web Consortium (W3C) is a body which promotes the standardisation of web design, especially of HTML. It aims to ensure universal accessibility of the web. For example, they promote the need for all websites to be displayed on a variety of browsers and resolutions so that they are usable by people with special needs. The W3C produces guidelines and tools for standardising websites which contribute towards increased accessibility. HTML compliance plays an important factor in this because if we use outdated HTML coding techniques or conventions this prevents a website from becoming fully accessible.

Link

For more on the W3C and their Web Accessibility Initiative visit their website: **http://www.w3.org/WAI/**

Accessibility features

- Alternative tags (alt tag) – An alt tag is a text alternative for an image or object on your web page (see Figure 15.12). If the image cannot be displayed the alt tag will be read in its place. The alt tag will be read by screen readers and other website readers as an alternative to the image itself. This is a necessity for blind users who use the internet, as the screen reader will tell the blind user what the image is, by reading the alt tag aloud. Therefore alt tags need to be meaningful: that is, say what is shown in the image in the context of the text on the page.

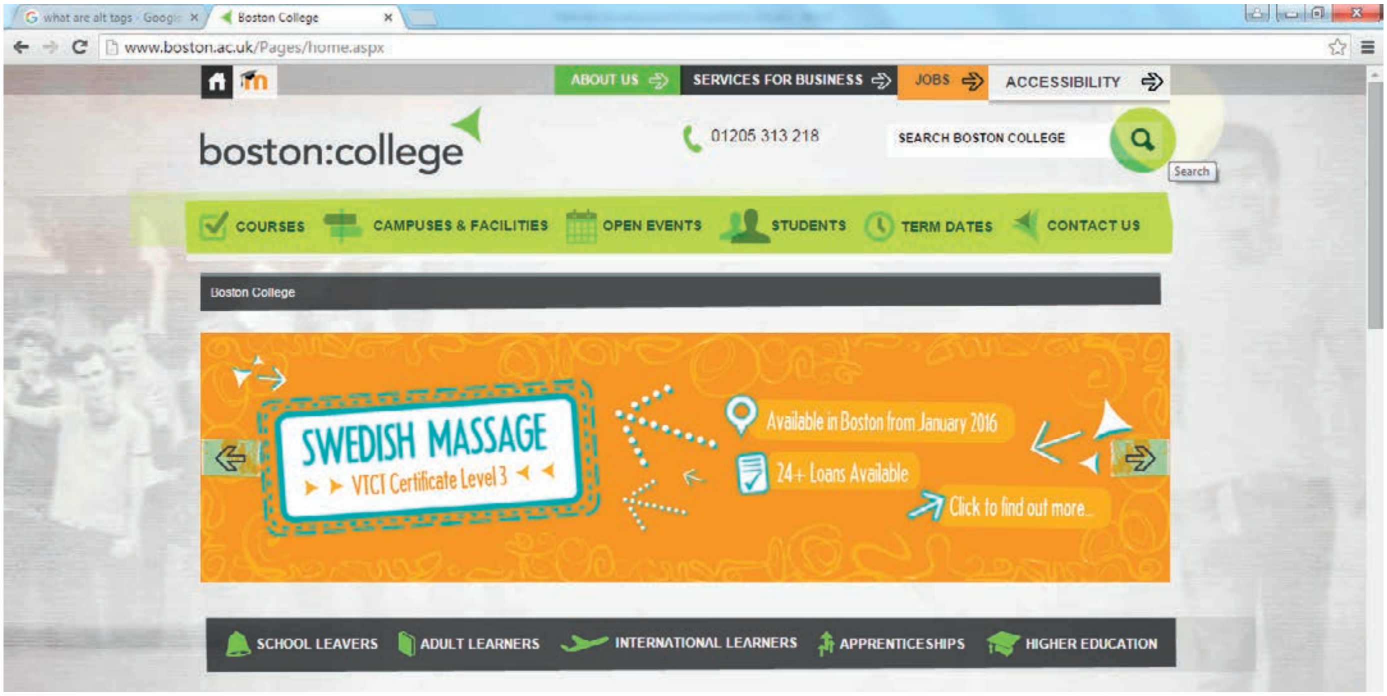


Figure 15.12: As you can see by hovering over the magnifying glass, the title tag is clearly stated as 'Search'. This means that a screen reader will be able to read this and a blind person will know what this icon is.

- Zoom features – Web browsers have a feature which allows the user to zoom in or out, making the content on the screen larger or smaller depending on their preference. This is particularly useful for users who have poor vision.
- Text-to-speech – For visually impaired users, text-to-speech (TTS) is very useful. It is a type of speech synthesis application that is used to create a spoken sound version of the text in a web page.

Platform compatibility

When developing websites, it is important that they are tested across a range of different platforms, particularly if it is part of the client's brief that the website should be usable on these platforms. These platforms can be broken into three categories:

- web browsers
- operating systems
- mobile devices.

When testing a website, you must ensure that it has a consistent appearance across these different platforms. The consequence of not doing this is that your website may display incorrectly on one or more web browsers using a particular operating system, or on certain mobile devices. Consider that every user visiting your website may use a different combination of browser and operating system, or be using a mobile device. Therefore, you must cater for all platforms and test your website to ensure that it appears the same across them all.

Obviously, developing a website to be compatible across multiple platforms is expensive, so this is an important constraint and consideration for clients and website developers. It might be that, initially, a client decides to focus on ensuring compatibility with only the most popular web browsers and the most popular mobile device operating systems, with the intention of ensuring compatibility with the others once the website has been established successfully on the most popular ones.

Embedding and compression of assets into suitable file types

Compression is a way of making a file smaller so that it uses less disk space. By using smaller file types that use compression methods, the website will have a faster download time making it more user-friendly. When deciding on which file types to use, a website developer must make a judgement in order to balance quality and file size because the higher the quality, the larger the file size. Once you have put your assets into the appropriate file types, they can be embedded within your web page. However, consideration also needs to be given as to what type of asset you are going to select.

Image files

There are two image file types available: bitmap and vector.

Bitmap file types include .gif and .jpeg. A .gif file has a maximum palette of 256 colours, and should, therefore, be used when the quality of the colour images is not that high and for images that do not contain many shades of colour. A .jpeg has a larger colour palette and is, therefore, better for high-quality colour images that include a lot of shading.

For a comparison of bitmap and vector images, see Table 15.6.

Digital sound files

Sound travels in waves. Natural sound waves are continuous and analogue. Digital sound waves are sampled at regular intervals with gaps, which are so small that the human ear cannot detect them. Once these signals are combined, the whole piece is a series of waves that denote the characteristics of the sound. As a computer can only understand 0s and 1s, the value of each part of the wave is converted into a binary value, for example 0000, 0001, 0010, 0011. These values are then translated by the computer into sound output.

There are several types of sound file type available, each with its own method of sampling and compression. A .wav file has a high sample rate, which means that the sound quality is closest to that produced by actual instruments, but it has a relatively large file size. A .mp3 file tends to have a lower sample rate and therefore produces a smaller file size. (This is how MP3 players manage to store such a high volume of music.) However, there is a loss of quality with .mp3 files compared with .wav files, depending on the compression rate chosen. The higher the compression rate, the smaller the file size, but this is at the cost of some fidelity. Consideration must also be given to the types of music **plug-ins** that a user is likely to have, as this may restrict the choice of compression rate available.

Key term

Plug-in – Software that will play specific types of files. For example, modern versions of web browsers like Internet® Explorer® come with Flash® Player which is a plug-in to allow the user to play Flash® animations.

Digital video and animation files

Videos and animations can seriously affect the speed at which a website is able to load and, in general, should be used sparingly. Both video and animation file types can produce very large file sizes.

Traditionally for users to be able to view videos or animations embedded within a web page, they used to have to click on them and download them. Due to the size of the files, this would often take a relatively long time and control a large proportion of bandwidth during the download, even with a high bandwidth capacity. Files like this also take up a large proportion of web server space. However, with the advent of HTML5, more videos are being embedded into web pages which can now show animations and video without the need of a plug-in. Therefore, there is no longer the requirement for constant plug-in updates and this means that videos and animations will run much more quickly and seamlessly.

Exporting digital assets

Once your digital assets have been created and developed using an appropriate file type, these files can then be exported for website use. When exporting your files, you should develop a folder directory which will store all your assets including images, sounds, videos and animations, as well as CSS files. This will mean that all your files will be stored appropriately and you will be able to refer to them when you code your website solution.

Table 15.6: Comparison of bitmap and vector images

Feature	Bitmap	Vector
Nature of file type	Each pixel is saved individually with its location, colours and other details.	Coordinates of points and curves are saved as a mathematical equation.
File size	Generally has a large file size.	Generally has a small file size.
Resizing	The image will become pixelated.	The image will retain clarity.
File formats	.bmp, .gif, .jpg, .png	.eps.
Created by	Programs such as Microsoft® Paint and Adobe® Photoshop®.	Programs such as Adobe® Illustrator® and CorelDRAW®.
Used for	Usually used in web pages as they are rendered by all graphical browsers.	Often used for graphics such as logos which need to be resized.
Shapes	N/A	Drawn in Adobe® Flash®.

C Develop a website to meet client requirements

Once your website design is complete, when it has been tested, when all the assets have been prepared and when the client has approved it, the website can be built.

Website development

The first element that needs to be created to build an interactive website is the structure. This will provide a solid basis for the content, which can then be easily inserted. Extra features, such as interactivity and audio-visual elements, can then be added. Once the website has been built, all elements must then be tested to ensure that they are functioning correctly. Once the website developer is happy that there are no bugs in the website, it can be uploaded to a web server and go live on the internet. This section will cover the creation of interactive websites.

Using cascading style sheets and accessing CSS from HTML

Cascading style sheets (CSS) allow you to create a standard layout and style which can be used easily on each web page in the site. Due to this standardisation, it is also easier to alter and maintain the site. For example, in HTML to change the font colour of all the titles to green would involve changing each one individually. When using CSS, only one value would need to be changed, and the change would immediately be applied (cascaded) throughout the whole site for every title.

CSS can be written into the HTML in three different ways: inline, header and external.

Inline

The CSS is defined in the same area of the code as that to which it is to be applied.

inline.html

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:red; Font-family: Courier New; font-size: 40pt;">
This demonstrates an example of inline CSS Heading</h1>

</body>
</html>
```

Header

The CSS is defined in the head section of each web page and applied throughout the website.

header.html

```
<!DOCTYPE html>
<html>
<head>
<style>
H1 {
Background: red;
Color: white;
Font-family: Times New Roman;
}
</style>
</head>
<body>
<h1>An example of header CSS</h1>
</body>
</html>
```

External

The CSS is defined in a separate file, which all web pages can reference. This is a .css file, rather than a .html file. The line, which can be put in the head of the HTML to link to the external CSS pages, is shown in Figure 15.13.

external.html | styles.css

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This demonstrating what happens to the header tag</h1>
<p>This is demonstrating what happens to the paragraph tag</p>

</body>
</html>
```

external.html | styles.css

```
body {
    background-color: grey;
}

h1 {
    color: green;
    font-size: 25pt;
    font-style: italic;
    font-weight: bold;
    font-family: Arial;
    text-align: center;


}

p {
    color:white;
}
```

See the examples in Figure 15.13 and Figure 15.14.

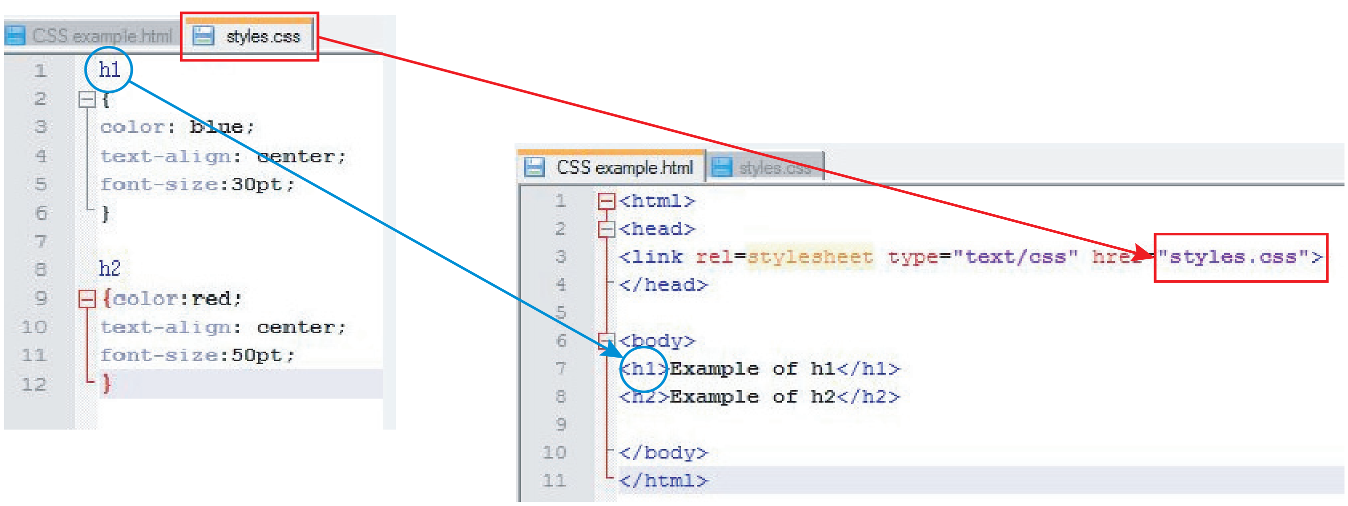


Figure 15.13: Example 1 of external CSS being applied

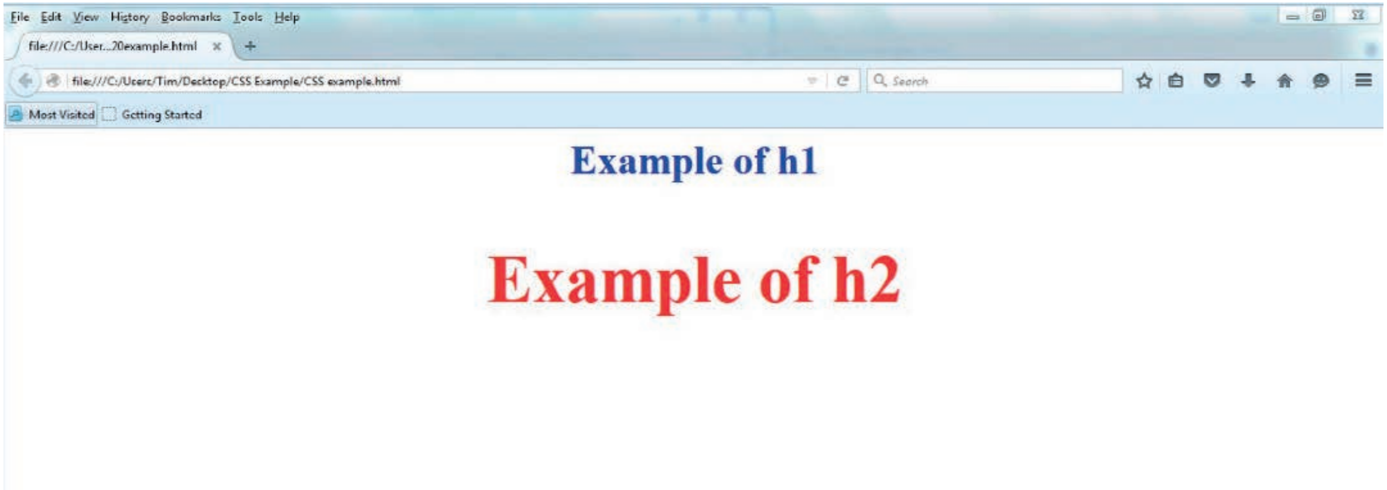


Figure 15.14: Example 2 of external CSS being applied

Other features of CSS

CSS can be used to alter the layout and formatting of any web page. Here are some examples of the hundreds of properties that can be changed.

These properties use the external method of applying CSS.

- Changing the background colour:

Background image.html | styles.css

```
Body
{
    Background-color:yellow;
}
```

- Changing the background image:

Background image.html | styles.css

```
Body
{
Background-image: url('picture.jpg');}
```

- Formatting text:

Formatting Text.html | styles.css

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>How to format text using CSS</h1>


</body>
</html>
```

Formatting Text.html | styles.css

```
H1
{
Font-family:serif;
Font-style: italic;
Font-weight: bold;
Font-size: 50px;
color:red;
}
```

These properties use the header method of applying CSS.

- Applying borders:

Border.html

```
<!DOCTYPE html>
<html>
<head>
<style>
p.border1 {
    border-style: groove;
    border-width: 5px;
}

p.border2 {
    border-style: double solid;
    border-width: medium;
}

</style>
</head>
<body>

<h1>How to apply borders</h1>

<p class="border1">Example of Border1</p>
<p class="border2">Example of Border2</p>

</body>
</html>
```

- Applying padding:

Padding.html

```
<!DOCTYPE html>
<html>
<head>
<style>
td {
    padding: 15px;
}
</style>
</head>
<body>
<h1>Table Padding</h1>
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Sex</th>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>Rowden</td>
    <td>Female</td>
  </tr>
  <tr>
    <td>Michael</td>
    <td>Bean</td>
    <td>Male</td>
  </tr>
  <tr>
    <td>Kelvin</td>
    <td>Andrew</td>
    <td>Male</td>
  </tr>
</table>
</body>
</html>
```

Step by step: Creating a simple CSS Page

4 Steps

1 Open Notepad++ or similar text editor. Save the file as myCSSwebpage.html.

2 Enter this code and then save the file.

```
<!DOCTYPE html>
<html>
<head>
<title>My First CSS</title>
<link rel="stylesheet" href="myCSS.css">
</head>
<body>
Here is some normal text.
<br>
<h1>Here is the text with CSS tags.</h1>
</body>
</html>
```

3 Create another new file and save it as myCSS.css. Enter the following code and then save the file.

```
h1 {
font-family: Arial;
color: red;
font-size: 20pt;
}

```

4 Navigate to where the .html file is saved using My Computer. Then double click your .html file to open in your browser.

PAUSE POINT

Following on from the Step by step, complete the following.

1. In myCSS.css, change the font to Wingdings. Save the .css file and refresh the .html file in the browser.
2. In myCSS.css, change the colour to blue and the size to 100pt. Save the .css file and refresh the .html file in the browser.
3. Now that you have done this, save an image from the internet and embed this into your css.
4. Develop the website to have different alignment of content.

Hint Use the external method of CSS as it is easier to separate the HTML from the CSS. Sometimes, this makes development of your website easier.

Extend Consider how to position your elements on your web page so that it is not restricted to alignments.

Key term

Pixel perfect – is a term used in the design sector to describe graphics that are accurate to the very last pixel.

CSS box model

CSS is used to create layouts on web pages. Using this method, the pages can be viewed in any web browser or at any resolution and the integrity of the design should remain. This is because the layout is recalculated on each opening. The resulting web page can therefore be designed very accurately, and can be **pixel perfect**.

The CSS box model structures the web page in a similar way to a table. Margins, borders, padding and content are each defined (see Figure 15.15 for an example).

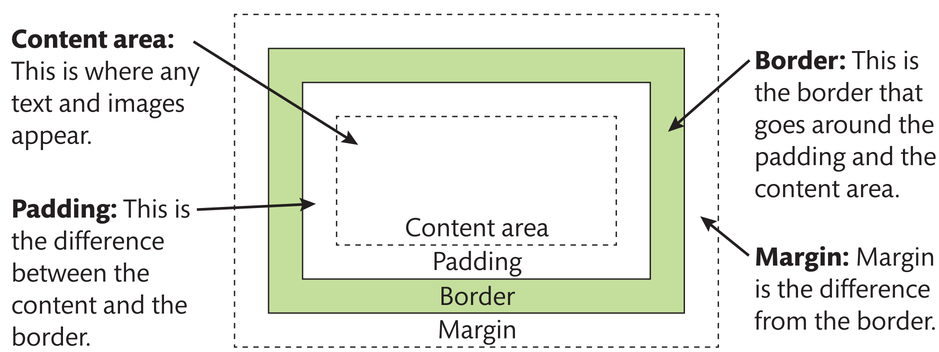


Figure 15.15: CSS box model

- Content area – Where the text and images which will be displayed on the web page should be placed. There can be more than one content area on a web page.
- Padding – The blank space around the content area, which ensures that the content is not displayed right up to the edges of the border.

- Border – The design surrounding the padding and content area, which defines the edge of the box.
- Margin – The blank space around everything so that the box does not display right up to the edges of the screen.

The padding, border and margin are optional and, if not defined, are set at a default value of zero. At this value they would be invisible.

HTML

It is important to realise that HTML evolves throughout time, and the most current up-to-date version of HTML is HTML5. In HTML5 there are new tags that have been introduced, and some tags that were present in previous versions have been removed.

Research

Research the differences between HTML 4.01 and HTML5. What are the new tags which have been brought in, which have been removed and what are the other differences? What do you feel the challenges are of using HTML5?

Table 15.7 describes some common HTML tags, their purpose and examples.

Table 15.7: Common HTML tags

Open tag	Close tag	Purpose	Example
		Changes text. Open tag can have parameters such as colour, size, face.	Text
		Makes text bold.	Text
• 		Makes text italic.	Text
		Creates a list with bullet points.	- first item - second item
<table>	</table>	Creates a table (<tr> creates rows and <td> creates columns).	<table border=1> <tr> <td>top left</td> <td>top right</td> </tr> <td>bottom left</td> <td>bottom right</td> </tr> </table>
<img>	No close tag	Inserts an image. One of the rare tags that is not in a pair.	<img src=”Images\logo.gif”>
<a href>		Creates a hyperlink. Can be used around text or an image.	Go to home page <img src=”Images\logo.gif”>

Doc types

The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag. This is not an HTML tag. Instead it is an instruction to the web browser about what version of HTML the page is written in. When creating a web page you should always add the <!DOCTYPE> declaration to your HTML documents, so that the browser knows what type of document to expect. When using a rapid application package (RAD) such as Adobe® Dreamweaver®, this automatically puts it in for you. When coding a web page using a text editor, you will have to put this in manually.

The most up-to-date doc type is in HTML5 and looks as follows.

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
The content of the document......
</body>

</html>
```

Client-side scripting

Even though HTML is the basis of all web pages, as a language it is quite limited and so other languages need to be brought in to create more advanced features on web pages. A client-side scripting language is used to write code that is embedded into the HTML. When the web page is downloaded onto the user's browser, the script is run on the user's computer.

Link

To remind yourself of what client-side scripting is, see Where scripting runs.

VBScript® and JavaScript® are client-side web languages. This means that the code is executed using the user's computer and not the web server. This frees up the processing power which would otherwise have been used on the server. Both languages can create interaction on a website, for example forms, searching and even games. Although VBScript® and JavaScript® are used to create extra functionality within web pages, they are different and have different uses. VBScript® is a simplified version of Visual Basic® that Microsoft® developed to deal with the static nature of websites. The disadvantage of using VBScript® is that it is only supported in Internet Explorer®, whereas JavaScript® offers cross-platform support for most web browsers. Table 15.8 below denotes some of the differences between the two languages.

Table 15.8

JavaScript®	VBScript®
Tends to be the default scripting language for most web browsers	Not the default language of choice by website developers
Offers cross-platform support for nearly all web browsers	Supports Internet Explorer® only
(+) is used for addition as well as concatenation	& is used for concatenation
Case sensitive	Not case sensitive
{ } are used to denote functions	Uses Function and End Function

Embedding client-side script into a web page

Follow the process below to embed original client-side scripts into web pages to provide more interactivity and improve the usability of a website.

Step by step: Creating a simple catalogue search using JavaScript®

4 Steps

1 Using Notepad++ or a RAD tool such as Dreamweaver®, create a catalogue web page with three products.

2 Below the <body> tag, enter the following code.

```
<!DOCTYPE html>
<html>
<body>
<script>
necklace = 1
chocolates = 2
toy = 3
product=prompt("Please enter search product", "")
if (product=="necklace")
{document.write("Item found. Catalogue number " + necklace)}
else
if (product=="chocolates")
{document.write("Item found. Catalogue number " + chocolates)}
else
{document.write("Item found. Catalogue number " + toy)}
</script>

</body>
</html>
```

3 Amend your code to match your three products.

4 Run the web page in a browser to test if it works for all three products.

PAUSE POINT

Client-side scripting can be used to provide different kinds of interactivity for web pages. Using resources such as the internet or books, see if you can create the following functionality within a web page.

1. Place today's date in the header of a web page.
2. Place the current time in the footer of a web page and make it work in real time.
3. Using a form text field, enable it so that when you type in text and press the <Tab> key it will convert the text to uppercase.
4. Display an alert when someone visits your website.
5. Create a hit counter.

Hint Consider working in pairs to see if you can come up with the correct solution. Website developers often work together to develop solutions to problems.

Extend Try to embed the JavaScript® coding into an external file (this is similar to the way in which you would create a CSS external file). Research how this would be done, and what advantages it would have.

Use of scripting languages

A scripting language such as JavaScript® has many uses. Some of the uses of JavaScript® are outlined below.

Alerts

Pop-ups to alert the user to something.

```
<!DOCTYPE html>
<html>
<body>

<script>
alert('This is what an alert message looks like.');
</script>

</body>
</html>
```

Confirming choices

Confirming choices is a form of **validation**. This gives a user the opportunity to check if they are sure that they clicked the correct option.

```
<!DOCTYPE html>
<html>
<body>

<button onclick="confirmChoice()">Click and see what happens</button>
<script>
function confirmChoice() {
    confirm("Press a button");
}
</script>

</body>
</html>
```

Key term

Validation – an automatic computer check which ensures that data entered is sensible and reasonable.

Browser detection

Used to determine what browser you are using.

```
<html>
<body>

<script>
document.getElementById("demo").innerHTML =
"Name is " + navigator.appName + ". Code name is " + navigator.appCodeName;
</script>

</body>
</html>
```

Creating rollovers

Rollovers add more visual interactivity. A web page can use rollover images or, in this case, buttons.

```
<html>
<body>

<img onmouseover="makeBigger(this)" onmouseout="normalImage(this)" border="1" src="lobster.gif" alt="lobster" width="50" height="50">
<script>
function makeBigger(x) {
    x.style.height = "80px";
    x.style.width = "80px";
}
    function normalImage(x) {
    x.style.height = "32px";
    x.style.width = "32px";
}
  </script>

</body>
</html>
```

Handling forms

Handling forms allow users to fill in forms and submit them, either for the website to process or by email to an inbox. This example enables a user to disable and enable a drop-down list using JavaScript®.

```
<html>
<head>

<script>
function disablecombobox() {
    document.getElementById("selectFish").disabled=true;
}
function enablecombobox() {
    document.getElementById("selectFish").disabled=false;
}
</script>
</head>
<body>

<form>
<select id="selectFish">
  <option>Shellfish</option>
  <option>Monkfish</option>
  <option>Skate</option>
  <option>Prawns</option>
</select>

<input type="button" onclick="disablecombobox()" value="Disable list">
<input type="button" onclick="enablecombobox()" value="Enable list">
</form>

</body>
</html>
```

Validating input

Validating input is a technique used to see if a user has entered text within a textbox. Here is an example.

```
<html>
<head>

<script>
function validateEntry() {
    var x = document.forms["validation"]["forename"].value;
    if (x == null || x == "") {
        alert("You cannot have a blank field.  Please type in your forename");
        return false;
    }
}
</script>
</head>
<body>

<form name="validation" action="validation_blank_field.asp"
onsubmit="return validateEntry()" method="post">
Forename: <input type="text" name="forename">
<input type="submit" value="Submit">
</form>

</body>
</html>
```

Constructs

Constructs are syntactically permissible parts of a program, and must be used in accordance with the rules of the programming language you are using. As with any code, scripting languages need to use the correct construction in order to work. This includes the **syntax**. It is important for any programming language that the syntax is correct. In the example below, the **dot operator** is used to allow an object to use a method.

```
<html>
<body>

<script>
str="First Message"
document.write(str.replace(/First/,"Second"))
</script>

</body>
</html>
```

Key terms

Syntax – a set of rules that is unique to each programming language, which defines the combination of symbols considered to be correctly structured within that language.

Dot operator – is a full stop (.) used to define what method an object will use.

Array – a collection of indexed variables, each of which has a single value.

Here is a list of various constructs.

- Loops – Loops are useful if you want to run the same code repeatedly, each time with a different value. Loops can often be used with **arrays**.
- Decision making – This is the process of using a statement whereby the user is forced to make a choice on a web page. For example, are you sure you want to exit this page? The options being either 'Yes' or 'No'.
- Functions – A function is a block of code designed to perform a particular task . This can be executed when something calls it into action.

```
<!DOCTYPE html>
<html>
<body>
<script>
function hello()          <!--This is where the function is named-->
{
alert("Hello World!")
}
</script>

<input type="button"
onclick="hello()"      <!--This is where the function is called-->
value="Click here!">

</body>
</html>
```

- Parameter passing – Parameter passing occurs when a value is passed to a function and then the function uses it while it is running.
- Handling events – Events are 'things' that happen based upon user interaction or something that the browser does. For example:
  - an HTML web page has finished loading
  - an HTML input field was changed
  - an HTML button was clicked.
- Methods – A method is an action that can be performed by an **object.** The image below shows how the method UpperCase is used to force the whole word to be in upper case.

```
<!DOCTYPE html>
<html>
<body>
<script>
hw="Hello World"
document.write(hw.toUpperCase())
</script>

</body>
</html>
```

Key term

Object – an object is a type of data that knows things about itself (its properties) and knows how to do things (methods).

Other issues involved in website development

Once you have developed your website using client-side scripting languages, there are a few other things you need to consider before your website development will be complete.

Compatibility with mobile and tablet devices

It is important when you develop a website that you consider how to make your website compatible with mobile and tablet devices. This is called **responsive web design (RWD)**. RWD involves using CSS and HTML5 to resize, hide, shrink, enlarge or move the content to make it look good on any screen. There are a number of methods of RWD for making a website compatible with mobile and tablet devices.

Key term

Responsive web design (RWD) – makes your web pages appear correctly (look good) on all types of device, including desktop PCs, mobile and tablet devices.

Effective use of tools and techniques

There are many scripting languages, such as HTML, JavaScript®, CSS and it can be quite overwhelming to use the techniques associated with using them to produce websites. There are additional tools available that can help you to produce websites more easily, for example the rapid application development (RAD) package Adobe® Dreamweaver®. These RAD packages allow you to create websites very quickly. For example, if you wanted to centre some text you would simply highlight the text and click on the central alignment button. The code for this change would automatically be generated for you. This may seem great and more effective to use. However, you do have more control over your website when you code it manually using a text editor.

Uploading of files to a web server

To allow a website to be seen by the public across the internet, it must be uploaded on to a web server (going live). The process of uploading involves a protocol called File Transfer Protocol (FTP). Uploading files via FTP (commonly known as FTPing) can be done directly through a web browser or by using a program such as FileZilla®.

It is not only the web pages that must be uploaded onto the web server, but all the associated files including media assets and CSS files. This is because these files are not embedded into the web pages. Instead, they are linked to them but remain as separate entities.

Reflect

Working to time and schedule is important when developing a website. Timeframes are often defined at the outset of a project. Someone will need to take responsibility for ensuring that all parts of the project are delivered on time. For example, set amounts of time will be given to the requirements stage, the design, the coding, testing and so on. Strong leadership is important to oversee all of these stages. If one of the stages takes too long, or is incorrect, then this has a knock-on effect on the following stages. Therefore it is important, when managing a website project, that there is strong leadership and good communication.

Website review

After your website is built, it is essential to review it to ensure that it is suitable for its intended purpose and audience, and meets all the client's requirements. This can also identify any areas for further improvement.

Quality in comparison with other similar websites

Once your website has been completed, it is useful to compare your website against similarly themed websites. Remember that millions of websites are created daily, so you have to do more than your competitors to make your website stand out. Comparing your website against similar websites will enable you to identify areas where your website stands out positively (compares favourably). There may also be elements of other websites that are better than yours or that work particularly well which could be incorporated into your own. You can take ideas from these websites to use for future improvement of your website. You might also identify areas in need of improvement in the other websites. Therefore, when further developing your website, you can use strong features from similar websites and leave out those features which you have identified as in need of improvement. This research will help you to produce a website which will stand up to competition from similar websites.

Suitability for intended purpose and audience

Your website will need to be reviewed to establish whether or not it is suitable for the purpose and audience for which it was intended. Quite often this will involve carrying out some form of market research to get feedback from potential users and discussions with the client to see if they are satisfied with the overall website. Remember that, while you may have been hired to develop a website for your client, ultimately it is other people that will be using it and these are the people that you (as well as the client) need to cater for. It is possible that the client might not fully understand the design preferences of the intended audience of the website, in which case your user feedback may help to inform both you and the client about how the website could be improved.

Suitability against the client requirements

During the beginning of the website development lifecycle, one of the first elements to be generated was a list of requirements. These SMART requirements state what the website must be able to do and how it should work. In order to know if the website development has been successful, it is important to compare the original requirements against the final developed website. There will need to be an appraisal to establish whether or not you have been successful in fulfilling the original requirements. If you find that there are some requirements that have not been fulfilled, then these areas can be optimised and developed further to meet the client requirements fully.

Reflect

Taking individual responsibility is a crucial behaviour attribute, whether in the IT industry or any other walk of life. There are always times when we fail to meet expectations. This is human nature and there is nothing wrong with it as long as we take responsibility for our own actions (we are accountable for them). The most important thing is that, when this does happen, a person understands what went wrong and puts measures in place to ensure that it does not happen again.

Legal and ethical constraints

Your website will need to be reviewed to ensure that it complies with any legal and ethical considerations. For example, it has already been mentioned that websites must be fully accessible. This means that your website must have elements such as:

- having alternative tags embedded within images
- having clear navigation
- ensuring documents/web pages are understandable
- not using colour alone to provide meaning.

Moreover, your website will need to be reviewed to ensure that it does not break current legislative laws, in particular the Data Protection Act 1998 and the Copyright, Designs and Patents Act 1988.

Strengths and improvements

You may notice that websites evolve and change with the times. Good websites recognise that they have to stay up to date with the latest trends in good design and cater for the needs of all users. Therefore, once your website has been developed, it is important to identify areas of strength and any areas in need of improvement within your website.

Consider a social networking website such as Facebook Inc which has millions of registered users. The Facebook Inc website has continuously evolved, considering feedback from others and making improvements. As a website developer, you will seldom be satisfied with the end product and should always try to find ways of improving and adapting your website.

However, it should be noted that there are limits on the improvements you can make to websites. Web developers cannot always take on board all comments and feedback because they may be constrained by time and budget. Therefore you need to carefully select which improvements can be made. These decisions need to be considered within the website development team and with the client. You should consider which improvements will be of the greatest benefit to the website and which will be most beneficial to the client's business. Other improvements should be set aside for now but be planned as part of future development at a later date.

Website optimisation

As well as reviewing the website, it is essential that the website is fully tested to ensure that it is fit for purpose and works correctly. There may be elements within the website which do not work as intended and therefore need to be optimised in order to fulfil the client's requirements.

Your website can be optimised so that it runs more quickly and efficiently. The following are ways in which your website can be optimised to run much more effectively.

- Reduce HTTP requests:
  - this can be done by using CSS instead of images, wherever possible
  - combine multiple style sheets into one
  - reduce scripts that run on the page.
- Compress large web files:
  - compression reduces the bandwidth used by your web page, thereby reducing the HTTP response time; there are online tools which enable you to do this.
- Avoid WYSIWYG resources:
  - WYSIWYG (**W**hat **Y**ou **S**ee **I**s **W**hat **Y**ou **G**et) are website resources that enable you to create a website quickly by inserting website objects; although they make it easy to build a web page, they do create messy code which can slow down your website considerably.
- Optimise Images:
  - oversized images can take longer to load, so it is best to optimise your images to the required size.
- CSS Delivery:
  - an external style sheet is the best method to optimise a website to its full potential; as only one external stylesheet is required, it reduces the size of your code and creates less code duplication.

These are additional areas of website optimisation that a user can test for and which would increase the performance of your website.

Performance and user testing

Performance and user testing is perhaps one of the most crucial forms of testing. This is used to test the functionality of a website and to ensure that everything works as expected. It is at this stage that you can use the test plan that was developed during the design stage and extend it to show the actual results and comments.

Link

Look back at Testing plan and, in particular, Table 15.5 Example test plan. You should use your test plan for user testing to show the actual results.

Tip

Remember to develop your test plan in a landscape orientation. This will give you more room to complete your testing and keep all related information together on one row.

Table 15.9: Example test plan with results

Test number	Purpose of test	On page	Test data	Expected result	Actual result	Comment/screenshot
1	Test the home button	About.html	Left click	Load Index.html	SUCCESS	It worked as expected. No further action required.
2	Logo.gif	Index.html	Load page	Appears in the centre of the page	SUCCESS	The logo appeared in the centre of the page as expected. No further action required.
3	Test the alt/title tag of Logo.gif	Index.html	Hover over logo	Tool text tip appears	FAILURE	It failed to work. When I hovered over the logo, nothing appeared. Screenshot 1
3B	Test the alt/title tag of Logo.gif	Index.html	Hover over logo	Tool text tip appears	SUCCESS	On this attempt it worked. I failed to put in the speech marks of the alt tag. Now works as expected. Screen shot 2

Remember that your test plan can also be used to check interactivity and compatibility with other web browsers. For example, if you have included any client-side scripting, such as including the time and date, then this should be tested. Remember that your website will be viewed in different web browsers. Consequently, the way in which your website may appear in one web browser may not be the same in another. Therefore, you will need to ensure that you complete your test plan in two or three different web browsers. When something does not work as expected, then this should be commented upon in the 'Comments/screenshot' column.

Tip

The Snipping Tool in Windows® is really useful for snipping images of your screen to show your screenshots of the website, whenever you have a test success or failure. To access it, go to the Windows Start screen, click on All apps, scroll down and click on Windows Accessories, then scroll down and click on the Snipping Tool.

Obtaining and evaluating feedback from others

Testing can take many different forms. As in the design stage, you could develop questionnaires to elicit feedback from others to help you refine your initial designs. You should obtain feedback from potential users of the website, in particular.

It can be hard to be critical about something which you, yourself, have developed. Therefore it is important that, when your website has been developed, you obtain feedback from your client, potential users and peers to see if the website works correctly, whether it meets the client's requirements and whether it is suitable for its intended purpose and audience.

Your initial design questionnaires can be adapted to elicit user feedback about the developed website using quantitative and qualitative questions. The feedback that you get will help you to identify where improvements can be made to your website. Be aware that you need to evaluate the feedback you receive about your website. Hopefully, most of it will be relevant and useful but it is possible that some of it is inappropriate or not helpful. For example, a client might feed back that the website does not have a particular element that was not in the brief. It is unreasonable and outside the scope of the website requirements for them to request this feature at this late stage. However, you may need to consider developing this feature if the client is willing to wait for the website and to pay for additional features.

Theory into practice

When asking your client and potential users for feedback on your website, it is likely that not all of the feedback will be positive. As an IT professional, it is necessary to have a 'thick skin' while also remaining objective and being professional. Remember that within the computing industry reputation is very important. Therefore, understanding how to respond to outcomes and how to communicate effectively are key attributes of a website developer.

Once your website has been completed, you will need to present and launch the website that you have developed. You will need to provide clear and comprehensive feedback on the product, and show how it fulfils the original requirements that were defined in the design stage.

1. Consider how you will be positive in the face of negative criticisms of your website. List three things that you would do to deal with negative feedback in a constructive way.
2. Consider how you will effectively present your completed website to the client. Summarise what tools you will use to present your website and what you will include in the presentation.

Reflect

Once the website development has been completed, you will need to provide a justification and rationale of the decisions made. It is important that when doing this you refer back to the original requirements and the design documentation. By carrying out an evaluation of the outcome of your work, you can convey to the client that a high-quality product has been developed because you are able to justify your decisions based on all the processes that have been undertaken.

Carry out an evaluation of your website development outcomes.

Assessment practice 15.2

B.P2 B.P3 C.P4 C.P5 C.P6 B.M2 BC.D2 C.M3 BC.D2 BC.D3

You were successful in the first part of your interview. The recruitment manager was very impressed by your report on the principles of website design. As a result, you have been placed on probation and given a twelve-week trial, where you have been asked to work on a website project.

You will be responsible for the design, development and testing of the website. It is hoped that, if you do a satisfactory job, the recruitment manager will authorise you to pass your probation and provide you with a full-time job.

Design stage

Your client is a local county council who want a website to advertise a town. The website that you produce will be used to promote the area and improve tourism. Therefore you will need to highlight the positive aspects of the town so as to attract visitors to the website.

A set of client requirements will be provided. You will need to produce design documentation including mood boards, storyboards, wireframes and flowcharts (to show any client-side scripting). You will work with a peer to review your designs and identify any areas in need of improvement. You will need to document the review that your peer completes for you so you will need to develop a questionnaire which you can give to them to elicit their feedback. It is important that the questionnaires ask a variety of questions so as to highlight areas of strength in the designs and identify those areas in need of improvement.

Be sure to include a completed test plan which you can use to test your developed website. This will need to be populated with a minimum of 20 tests.

After your designs have been reviewed and any improvements made, you will need to fully evaluate the design and justify the decisions that you made, explaining how it would meet the needs of the client.

Development stage

Using your design documentation, you are to produce a website which will fulfil the client requirements. As you develop your website, make sure that you test it as you go along. Be sure to use the test plan that you completed.

Once you have completed your website, review it by getting your peers to critique it. Therefore, you need to use questionnaires to gauge their thoughts and opinions. Be sure to optimise your website based on their feedback, by making the changes that they suggest (assuming they are good ideas). It is recommended that you keep before and after versions, annotating where the changes have taken place.

Finally, you should evaluate both the design and the final website that has been developed. This will mean looking at the overall process undertaken and identifying what went well and what did not go so well.

Throughout the whole process, you will also need to demonstrate individual responsibility, creativity and effective self-management. This is important, as the recruitment manager is eager to know that you have the necessary traits of being able to work responsibly, professionally and under pressure to fulfil the role of a web developer.

Plan

- What is the task? What am I being asked to do?
- How confident do I feel in my own abilities to complete this task?
- Are there any areas I think I may struggle with?

Do

- Have I spent some time planning my approach to the task?
- Am I confident that I know what I am doing and that I know what it is I should be achieving?

Review

- I can explain what the task was and how I approached it.
- I can explain how I would approach the hard elements differently next time (ie what I would do differently).

Further reading and resources

Flanagan, D. (2011). JavaScript: *The Definitive Guide (Definitive Guides), Sixth Edition*. Cambridge: O'Reilly Media.

McFarland, D. (2015). *CSS: The Missing Manual, Fourth Edition*. Cambridge: O'Reilly Media.

McGrath, M. (2011). *HTML5 in Easy Steps, Seventh Edition*. Southam: In Easy Steps Limited.

Websites

www.csszengarden.com/
This website allows anyone to explore different CSS templates which can be applied to a website design.

www.webpagesthatsuck.com/
This website analyses well and poorly designed websites.

https://validator.w3.org/
The Markup Validation Service (W3C) allows you to validate website content for free. This website enables you to check for errors and ensure that your website is W3C compliant.

www.codecademy.com/
Includes free videos and training tutorials on how to develop websites.

www.w3schools.com/
A useful starting point for anyone who wishes to learn how to use HTML, CSS and JavaScript to produce websites.

THINK FUTURE

Michael Bean

Junior Website Developer

I've been working as a website developer for over three years now. I was fortunate to take an apprenticeship position with a company that hired me to learn on the job while I undertook my BTEC National in IT. The experience I gained while working at college and in the workplace has been invaluable. I was so pleased to become a full-time employee with the company I am at, as it gives me opportunities to develop my creative and artistic skills when creating websites. As well as this, there is a challenging, problem-solving side to the job where I have to develop solutions to problems that occur. Sometimes, these problems can be quite simple to solve. Other times they need more thought but, as I am working within a team, there are people I can talk to, and together the problems quickly develop into solutions.

If there is one thing I would like to pass on to aspiring web developers, it would be to always remember to place the needs of the client first. If the final website does not suit their requirements or does not fit the audience, then the client is within their rights to reject it. The client's needs must be balanced with the users' needs too. The client needs a website that people will want to visit, which draws them in and grabs their attention. Balancing these needs can be challenging, at the best of times. However, when you develop a website from the very beginning to its completion, it is a very satisfying achievement, and one which gives me pride in what I have done.

Focusing your skills

Designing a website

It is important to be able to design a website before you go ahead and actually develop it.

- What are the main methods in gathering feedback from your client?
- What are the implications if you fail to gather the requirements for the website?
- As a website developer, you are put under constraints. What might these constraints be? And what effect could they have on the development of the website?
- Finally, once you have gathered all the requirements, what design methods should you employ? To whom should these designs be presented?

Creating a website

Once you have received all necessary feedback and the design stage is complete, you can go ahead with developing your website.

- What would you consider to be the best method of developing a website? Would you use a rapid application development package such as Dreamweaver® or would you use a text editor?
- If your testing revealed that your home page does not look right or render properly in another web browser, what steps would you take to resolve this issue?
- What steps should you take if your client does not like the initial look of the website?
- How would you upload the final website to the worldwide web?

Getting ready for assessment

Kelvin is working towards a BTEC National in Computing. He was given an assignment with the following title: 'How website principles are utilised to creative effective websites' for learning aim A. As part of this assignment, he was provided with a real-life scenario which he may encounter when he attends interviews for the role of a website developer. The assignment is based on a Word® report explaining how two websites utilise website development principles. Kelvin will also need to:

- give an explanation of how the principles embedded within his chosen websites meet the client requirements
- give a comprehensive evaluation of how website principles incorporated within his chosen websites have been utilised to create a creative, high performance website.

Kelvin shares his experience below.

How I got started

First, I collected all my notes on this topic and collated them into a folder, for easy reference. My first task was to select two websites which I could compare. This proved more challenging than I expected because I wanted to select a website which included the majority of the elements that I had learned about in my lessons.

Once I had selected the websites, I printed out screenshots of the web pages onto paper. I began to highlight areas where website principles had been incorporated. This helped me enormously when I began to bring it all together to produce my assignment. I also highlighted areas where website principles were not incorporated appropriately. I made brief notes, in the margin, of the implications this could have on website visitors.

How I brought it all together

I decided to use Microsoft® Word® and to use a sans serif font such as Arial. I had learned that sans serif fonts are easier to read, especially for people with learning difficulties such as dyslexia. I wrote a short introduction, identifying the two websites I was going to discuss in my report.

- I produced a screenshot of the home page of each website.
- I explained how each of the websites was suitable for their intended audience and purpose.
- I gave an evaluation of how website principles contributed to creating two websites which were highly effective and creative.

I ended with a brief conclusion giving my overall thoughts on the websites. This helped to summarise my findings.

What I learned from the experience

I found the task more challenging than I initially thought it would be. I began to research websites which gave me full coverage of all the principles I had learned about in lessons. I did spend too much time on this, and this meant that I had less time to complete the report. This meant that my evaluation was not as detailed as I would have liked. I feel that the report would have benefited from having more detail about the way in which website principles are used to create high performing websites.

I also chose two differently themed websites. With hindsight, I think it would have been better to choose two similarly themed websites. If I had chosen two similarly themed websites I believe comparing them would have been easier and would have felt more natural than comparing two completely different websites. I would also consider that both of the websites did not need to be great, as then there would have been more to compare about them.

Think about it

- Have you written a plan with timings so you can complete your assignment by the agreed submission date?
- Do you have notes on all the elements of website principles that you have been taught?
- Have you included screenshots to give extra clarification to any justifications you have given?
- Is your information written in your own words with quotations from books, journals and websites and is it referenced clearly?

Mobile Apps Development 17

Getting to know your unit

Assessment

You will be assessed by a series of assignments set by your tutor.

Mobile devices are everywhere and their impressive sales are primarily driven by the innovative apps that keep their users educated, informed and entertained. The mobile applications sector enjoys a thriving commercial ecosystem and it can be profitably targeted by small start-ups and larger corporations alike.

Developing a mobile application is typically the result of having a good initial idea or solving an essential problem; making that mobile application really successful comes from an appreciation of the intricacies of mobile devices, the various forms of functionality available and how apps are designed to be intuitively usable. The ability to create such applications is a highly sought skill and will help you, as a software developer, gain a competitive edge.

How you will be assessed

This unit will be assessed internally by two tasks set by your tutor. Throughout this unit, you will find assessment practice activities that will help you work towards your assessments. Completing these activities will not mean that you have achieved a particular grade, but you will have carried out useful research or preparation that will be relevant when it comes to your final assignment.

In order for you to complete the tasks in your assignments successfully, it is important to check that you have met all of the Pass grading criteria. You can do this as you work your way through the assignments.

If you are hoping to gain a Merit or Distinction, you should also make sure that you present the information in your assignments in the style that is required by the relevant assessment criteria. For example, Merit criteria require you to review and justify and Distinction criteria require you to evaluate.

The assignments set by your tutor will consist of a number of tasks designed to meet the criteria in the table. The first assignment is likely to consist primarily of a research-based written task that requires you to investigate mobile apps and mobile devices, while the second will include practical activities such as:

- designing a mobile app that utilises device functions
- developing a mobile app that utilises device functions.

Assessment criteria

This table shows what you must do in order to achieve a **Pass**, **Merit** or **Distinction** grade, and where you can find activities to help you.

Pass	Merit	Distinction
Learning aim A Investigate mobile apps and mobile devices		
A.P1 Explain how the purpose of a mobile app and the needs, preferences and characteristics of the user affect its design and the provided features. **Assessment practice 17.1**	**A.M1** Analyse how the implementation and design of mobile apps is affected by the intended user, current technologies and the purpose of the app. **Assessment practice 17.1**	**A.D1** Evaluate how the effectiveness of mobile app implementation and design are affected by the intended user, current technologies and the purpose of the app. **Assessment practice 17.1**
A.P2 Explain the impact of current technologies on the design and implementation of mobile apps. **Assessment practice 17.1**		
Learning aim B Design a mobile app that utilises device functions		
B.P3 Produce designs for a mobile app to meet identified requirements. **Assessment practice 17.2**	**B.M2** Justify how decisions made during the design process ensure the design for the app will meet identified requirements. **Assessment practice 17.2**	**BC.D2** Evaluate the design and optimised mobile app against client requirements. **Assessment practice 17.2**
B.P4 Review the mobile app designs with others to identify and inform refinements. **Assessment practice 17.2**		
Learning aim C Develop a mobile app that utilises device functions		
C.P5 Produce a mobile app that meets the design criteria. **Assessment practice 17.2**	**C.M3** Optimise a mobile app that meets the design criteria. **Assessment practice 17.2**	**BC.D3** Demonstrate individual responsibility, creativity and effective self-management in the design, development and review of a mobile app. **Assessment practice 17.2**
C.P6 Test a mobile app for functionality, usability, stability and performance. **Assessment practice 17.2**		
C.P7 Review the extent to which the mobile app meets the identified requirements. **Assessment practice 17.2**		

Getting Started

Developing mobile applications requires the knowledge and application of many different skills. Write down a list of the tools and techniques you think you will need in order to create innovative mobile applications. When you have completed this unit, see if you have missed any obvious skills, tools or technologies from this list.

A Investigate mobile apps and mobile devices

Before starting to develop mobile applications, it is important to gain an appreciation of the range of devices and functionality available. Many design decisions are shaped by the freedoms and constraints of the target devices. In this section, we will investigate different types of mobile app, the devices on which they work and the features, characteristics and options that will shape your development path.

Types of mobile apps

Mobile applications (more commonly known as mobile apps) are programs designed to work on smartphones, tablet PCs and emerging wearable technology such as watches and glasses. An app is typically developed as one of three general types outlined in Table 17.1.

Native and hybrid applications require specialist software (and sometimes hardware) in order to be developed successfully. For example, Apple® iPad® and iPhone® apps are normally created using Apple®-specific software and hardware which guarantees development standards and practices.

In direct contrast, web applications can be developed using simple text-editor tools such as Microsoft® Notepad and may work equally well from within web browser client software running on a desktop PC, games console or mobile device. By being so versatile, they reward greater investment (time, money etc.) by developers because the potential user base is much larger.

Generally speaking, when choosing which type of mobile app to develop, there is a trade-off between development time and cost, the required functionality and the level of user experience required.

Discussion

If you have a mobile device, you are likely to have encountered a wide variety of different mobile applications. Think about these apps, especially their appearance and functionally, and determine whether they are likely to be native, web or hybrid apps. Discuss your conclusions with your class.

Context of mobile apps

Mobile apps are created to fulfil many different user needs, and this defines their context and purpose. Each app is designed to provide the user with a specific tool, experience or enhancement that their basic mobile device does not already provide (or do well). The overall interface design, development challenges and actual use once installed on the mobile device is dependent on the type of app.

Table 17.1: The three types of mobile app

Native apps	Web apps	Hybrid apps
• Programmed for, and installed on, a specific mobile platform. • Can take advantage of all (permitted) functionality available in the device, e.g. location, camera, contacts. • Typically mimic manufacturer's 'official' apps, drawing on the user's prior experience for ease of use.	• Remote applications, typically running on a server. • User typically interacts with the application through a mobile device's web browser. • No application or data is typically installed on the device. • Access to device functionality is limited. • May visually mimic the devices's usual interface.	• Usually cross-platform compatible. • Typically uses website components that are 'wrapped' inside a native application. • Access to device functionality may be limited. • May be seen as a more cost-effective development approach, especially when multiple types of mobile device are being targeted.

Table 17.2: Common categories of app context and purpose

Context	Purpose and features	Examples
Locale	Apps designed to provide you with geographical information based on their current location (e.g. maps, GPS route finders, augmented reality (AR) experiences).	Google Maps™, Wikitude™, Yelp Monocle™, Google™ Sky, Aurasma™
Utility	Apps provided to help configure or maintain your device (e.g. file manager (adds or removes files), backup tools (backup to the cloud), system monitors).	Glary Utilities™, CCleaner™, WinZip™, AVG® AntiVirus for Android™
Productivity suites	Apps giving office-style functionality for word-processing, spreadsheets, slideshows, databases. They provide the user with facilities to write letters or reports, create presentations, calculate project costings etc.	Adobe® Acrobat®; Microsoft® Office®, Office 365® and OneNote®; Dropbox™
Immersive full screen	Apps using the screen display exclusively to fully draw you into their experience (e.g. games).	Temple Run 2™, Candy Crush Saga™, Angry Birds™, Fruit Ninja™, Minecraft: Pocket Edition™
Lifestyle	Apps designed to enrich your life (e.g. healthy recipes, DIY, interior design, gardening, travel). They may provide step-by-step visual instructions and narrative, 3D mock-ups and links to commercial websites.	Craftsy™, Gumtree™, TripAdvisor®, Etsy™, Evernote™, Google Translate™, eBay™
Social	Apps designed to connect you to other people, to aid communication and share ideas, events, pictures and videos.	Twitter™, Facebook™, Snapchat™, WhatsApp™, Pinterest™, Instagram™, WordPress™, Blogger™
Entertainment	Apps that provide media content (e.g. music players, video streaming, podcasts, e-book readers).	YouTube™, BBC iPlayer™, Spotify®, Shazam™, SoundCloud™, Kindle™, Audible™
Widgets	Apps that take up very little of the mobile device's screen display which provide quick access to live data or settings (e.g. news tickers, quick device settings, search facilities, calendar and appointments).	Feedly™, Flashlight apps, Todoist™, BBC News

Building apps for each of these categories poses different challenges to the mobile app developer.

In simple terms, the mobile app you design and its features will be determined by the task(s) that it must perform and the personal preferences and needs of its targeted user.

Identifying specific user needs can be difficult, but there are some key points and questions to remember.

- User needs – what functionality does the user need from the app? Which features are required?
- User preferences – how does the user want to use the app? How does the user want to interact with the app? How does the user want to navigate the app's functions and features? What visual style should the app offer?
- User characteristics – how should the app cater for different types of user, e.g. age of the user, technical expertise of the user, disabilities such as visual impairment or hearing difficulties, the physical environment in which the app is used, the breadth of user configuration that the app offers, the level of help and assistance the app provides?

Answering these types of question builds a profile of the app's intended users and helps to shape both its design and included features.

You have probably experienced occasions where you have downloaded two similar mobile apps that claim to do the same job; their design, effectiveness and usability will help you evaluate which to keep and which to remove.

Theory into practice

Podcasting has become a popular form of entertainment in the last few years and there are many different podcast apps available for the popular mobile platforms.

Select three different podcast apps for a selected mobile platform and investigate their design, functions, features, performance and usability.

Which one would you recommend to a friend? Explain and justify your reasoning.

Mobile device integration

Mobile devices come in many different shapes and sizes (smartphones, tablets, wearable technology). In addition, their features and functions also vary greatly. This is partly due to the fact that mobile manufacturers wish to distinguish their products in the marketplace and the integration of specific sensors (fingerprint security,

for example) can be an attractive selling point. When designing a mobile app, it is necessary to be aware of these different characteristics and the implications they may have for both design and development.

Mobile device functions

As a developer, you cannot guarantee that a mobile device has a particular function when designing and developing an application so it is recommended that you check the operating system to see if the function is available. However, key functions such as vibration, headphone and speaker output, touch screens, microphone and still and video cameras can generally be guaranteed across any modern mobile device, irrespective of the manufacturer.

Figure 17.1: Spirit Level mobile app using an orientation sensor

User interface

A user interface (often abbreviated to UI), is the combination of software and hardware that a user interacts with to perform set tasks. The software part of the UI includes the underlying operating system and the selected mobile app itself. The hardware of the UI is represented by input devices (such as touch screens, physical buttons, camera and microphone) and output devices (such as screen, speaker and vibration).

Mobile apps development is challenging because there are many limitations that the designer and programmer need to navigate successfully in order to create a rewarding UI interactive experience. These challenging characteristics include:

- limited (or potentially variable) screen size
- limited keyboard/keypad input mechanism
- limited processing power (although in modern smartphone devices this has rapidly improved).

The use of alternative input mechanisms such as voice, touch control, complex gestures (pinch, stretch and swipe) and physical effects (shake and tilt) require extensive thought before they are used in an app, in order to produce a user-friendly experience. When used well, for example in a sketching app that allows the user to shake the device to wipe the picture clean, the results are pleasing and the experience is highly intuitive. When designed poorly, for example controlling a nimble playable character using poorly judged touch screen controls, the results can be imprecise and frustrating for users.

Table 17.3: Common mobile device functions, what they are and how they are typically used

Mobile device function	What it is	How it is used
Accelerometer	A motion sensor measuring the force applied to the device on all three axes (*x*, *y* and *z*), excluding gravity.	Used to track motion such as shaking the device and tracking the difference between a user walking or jogging.
Magnetometer	A positional sensor measuring the ambient magnetic field.	Used to create a compass.
Thermometer	An environment sensor measuring ambient or internal temperature, typically in degrees Celsius (°C).	Used to measure room temperature or the temperature of the mobile device itself.
Barometer	An environment sensor measuring air pressure usually in either hectoPascal (hPa) or millibars (mbar).	Used to measure air pressure in a room. Also used to measure altitude, to make GPS more accurate when height is important.
Photometer	An environment sensor measuring the ambient light levels in lux (lx).	Used to auto-adjust screen brightness depending on external light conditions to improve readability and conserve power.
Orientation	A motion sensor calculating tilt and orientation (see Figure 17.1). This uses the accelerometer.	Used to change the orientation of an app's display when a device is changed from being held in a portrait to landscape manner, or to create a spirit level.
Global positioning system (GPS)	A positional sensor using triangulating satellites or cellular base stations to calculate a real-world location.	Used to track the physical location of the mobile device for providing location-sensitive information (e.g. nearest restaurant or petrol station) or services (finding friends, live maps and route finding).

Even devices in the same family (for example Apple® iPhone® or iPad®) may have slightly different UI options, so you should never really take any feature for granted without researching it properly.

Variable features may include the type of touch screen (**resistive** or **capacitive**-based, single or multi-touch), dedicated physical buttons or just a very small screen. Mobile app developers need to think about how their application will function within the constraints of different user interfaces (and on different hardware) in order to support their targeted users effectively.

Key terms

Resistive touch screens – use two layers (usually glass and plastic) covered in an electrical conductive material (usually Indium Tin Oxide (ITO)). The two layers are kept apart until a finger or stylus presses them together, which causes a localised change in electrical resistance. This type of touch screen is cheaper to manufacture than capacitive touch screens but is not very sensitive and cannot support multi-touch gestures.

Capacitive touch screens – use two spaced layers of glass that are both coated with minute ITO capacitors. When the user's finger touches the screen, it changes the screen's local electrostatic field. This type of touch screen is brighter and more sensitive than resistive touch screens, and supports multi-touch gestures. However, the structural complexity of the screen raises its production costs.

Operating system

Mobile devices are controlled by their operating systems (OS) in a similar fashion to notebooks and PCs. Similarly, just as notebooks and PCs offer different operating systems (Microsoft® Windows® operating system, Linux, Apple® OS X®), the majority of the mobile device marketplace is divided between those devices running the Google™ Android™ OS (see Figure 17.3) and those using Apple® iOS (see Figure 17.2).

Both mobile operating systems support similar functionality but the way in which a developer makes use of each function as their app runs is often quite different, and this can also vary between versions of a mobile device OS. For example, it is quite common for a new OS version to 'break' a mobile app that was previously working because it has changed some small detail or setting.

In addition, the programming language used to build the app for each OS is typically different and provides its own challenges, rewards, advantages and disadvantages.

Traditionally Android™ apps are written using Oracle®'s JavaScript (although C or C++ may also be used), while Apple® iOS apps are created using Objective-C.

Figure 17.2: Apple® iOS

Figure 17.3: Google™ Android™

Device permissions

Mobile app developers will often want to access particular data or functionality contained in a mobile device in order to make their apps more appealing to customers. Mobile device manufacturers take security and the integrity of their devices very seriously; therefore, apps often need to be granted permission to use data and functionality within a device through manual user intervention.

You may well have experienced this yourself. A pop-up prompt will appear asking if the mobile app you are using may access your location, read your contacts or phone status, and even access your network (see Figure 17.4). Some mobile apps, such as podcast players, may even need permission to download data if not connected to a wireless network, simply because mobile network data charges can be expensive.

As a mobile app developer, you should not plan for your app to rely on data or functionality that might not be granted by the user.

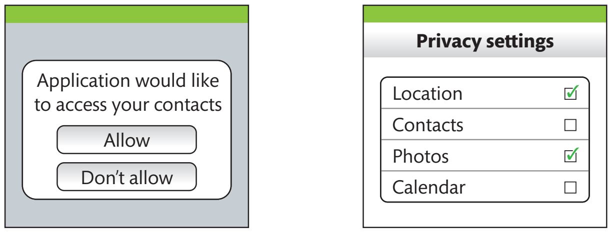


Figure 17.4: An operating system asking permission from the user for the app to access their Contacts information

Mobile app programming

As a mobile app developer your choices, including the programming language you must learn and the integrated development environment in which you build your application, are typically predetermined by the OS of the mobile device for which you intend to build your app, as shown in Figure 17.5.

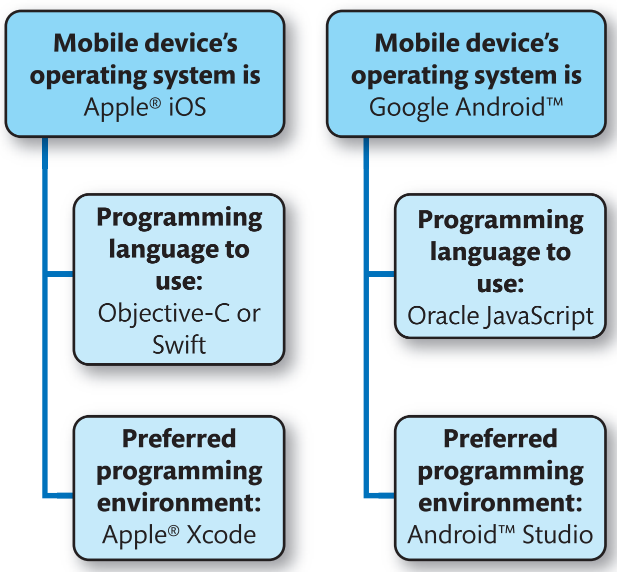


Figure 17.5: Determining the development tools and programming language

Programming languages

JavaScript is an established and popular programming language, which is over 20 years old and is used in billions of devices worldwide. The Android™ **software development kit (SDK)** uses JavaScript as the basis for building its mobile apps. Objective-C, a superset of an older language called C, has been around since 1983 and is the main language used by Apple® for its desktop OS X® and mobile iOS operating systems. Apple's Swift (2014) programming language, is seen as the preferred successor to Objective-C, being more concise and resilient to programming errors..

Despite being different programming languages, Objective-C and JavaScript are both **object-oriented programming languages** and are similar, in terms of their structure and syntax. As a result, you may find that becoming confident in one will be beneficial for learning the other.

Key terms

Software development kit (SDK) – a suite of software tools provided for developers to create an application for a specific platform, for example JavaScript SDK for Android™. The SDK usually contains application programming interfaces (APIs), tools such as compilers and debuggers, reference documentation and code samples.

Object-oriented programming languages – a modern programming approach (paradigm) to software development that works by modelling real-world problems and simplifies complex processes into the basic interactions that exist between different objects: for example, a customer and their bank account.

Developing for Android™ is essentially free, although a one-off $25 registration fee is required for publishing free or commercial apps that are to going be distributed through Google Play™. Apple® iOS development appears to be more expensive as it requires a $99 registration fee annually.

Research

Visit Apple®'s membership support page and find out about the various membership options that enable development of iOS apps for their mobile devices. To access this website go to: developer.apple.com

Learn how to register and publish Android™ apps through Google Play™ by visiting: developer.android.com

This book focuses primarily on Android™ and JavaScript, but you will find that doing research to make some comparisons is very useful.

Programming environments

The role of the modern programming environment is represented by an integrated development environment (IDE). The IDE provides developers with a comprehensive suite of tools, typically including:

- project management tools – to create, organise and manage your apps
- design tools – to create user interfaces, typically using 'drag and drop' techniques
- fully featured text editor – to key-in, edit and save your app's source code
- code completion – helpful auto-complete to speed-up development

- compiler – to translate your app's source code so it works on its target mobile device
- syntax highlighting – colouring different elements of program code to improve readability
- documentation tools – for adding comments explaining how your app works
- debug tools – to assist with identification and removal of program errors
- emulation tools – demonstrating your app running on a virtual mobile device
- testing tools – to show your app's performance, usage of resources etc.
- deployment tools – to transfer an app to its electronic store or physical mobile device.

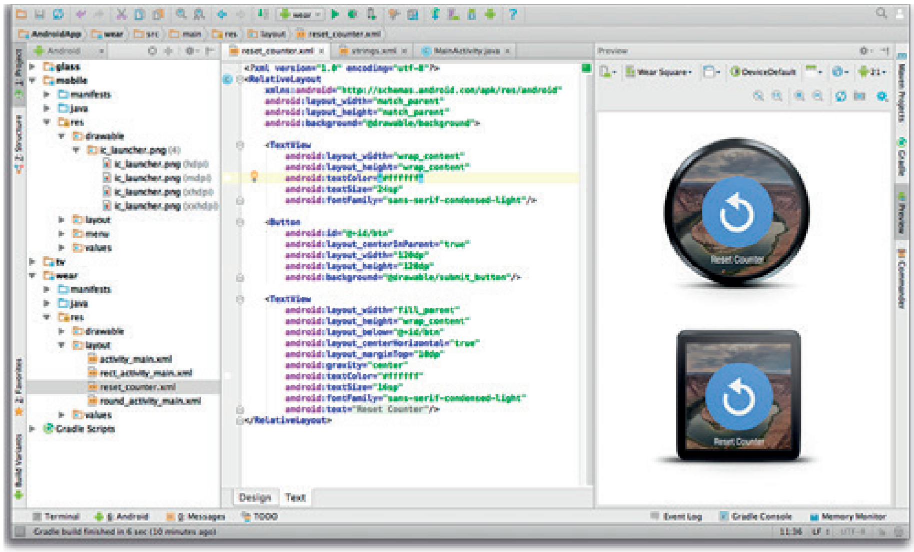


Figure 17.6: Android™ Studio is the official IDE for Android™ application development (Google and the Google logo are registered trademarks of Google Inc., used with permission)

PAUSE POINT Can you explain what the learning aim was about? What elements did you find easiest?

Hint Close the book and make a list of the different types and categories of mobile app available.

Extend Which factors affect the design and features of a mobile app?

Assessment practice 17.1

A.P1 A.P2 A.M1 A.D1

Following in your footsteps, another junior developer is joining your development team next week. Although they have some programming experience, they have very little experience of developing mobile applications but are expected to learn very quickly.

You have been asked by your line manager to prepare and deliver an induction presentation that explains how the design of a mobile application is affected by its intended purpose and the needs, preferences and characteristics of the target users. In addition, you have been asked to provide an overview of current mobile technologies, explaining how these affect the design and implementation of such apps.

You should conclude your presentation with a worked example that analyses the design and implementation of some sample mobile apps, given these influencing factors, and which evaluates how they might have an impact on the effectiveness of the design and implementation.

Plan

- What is the task? What am I being asked to do?
- How confident do I feel in my own abilities to complete this task?
- Are there any areas I think I may struggle with?

Do

- I know what I am doing and what I want to achieve.
- I can identify when I have gone wrong and adjust my thinking/approach to get myself back on course.

Review

- I can explain what the task was and how I approached the task.
- I can explain how I would approach the hard elements differently next time. (ie what I would do differently).

B Design a mobile app that utilises device functions

It is best practice to design your mobile application thoroughly before practical development takes place. In this section, we examine the design considerations and actions that you should take when designing your mobile application.

Analyse requirements for an app

Your first step when designing a mobile app is to consider the underlying computing requirements. Some of these are concerned with the capabilities of the device, while others focus on what the app needs to do (its core functionality) and the specific needs of the user. We can break this down using a quad diagram, which is a simple visual tool that focuses on four simple areas (see Figure 17.7).

It should be possible to complete a quad diagram for any prospective mobile app.

Device capabilities	Input required	Output required	User's needs
Functions such as accelerometer, GPS, tilt sensor, etc.	How user interacts with the app, e.g. voice, touch screen, timed events.	How the app provides feedback to the user, e.g. vibration, audio, video effects and transitions, etc.	Particular considerations, e.g. location-based services, accessibility issues, responsiveness, accuracy, etc.

Figure 17.7: Quad diagram for app requirements

Worked example: Traffic lights

A simple mobile app is needed which will teach the correct colour sequence of a UK traffic light (a mandatory aspect of the UK driving test). Users should be able to see a timed animation of the UK traffic light sequence and be able to play a simple game where they predict which light will be next in the sequence by touching the correct light. Correct responses should be greeted with a congratulatory round of applause; incorrect responses should make the device vibrate. Accessibility options could be included for users with colour vision deficiency to assist the learning process. The traffic lights are reset to the main menu via a simple shake gesture.

Step 1: Analyse and identify elements for each requirement category.

Step 2: Complete a quad diagram to organise your thinking.

Device capabilities	Input required	Output required	User's needs
Accelerometer to detect user shake action.	Touch screen for traffic light selection, menu options and navigation.	Feedback through animated visuals, sound effects (applause) and vibration.	Accessibility options for users with colour vision deficiency, e.g. different shapes (square, circle, triangle) for each light.

Step 3: Check that all elements have been correctly identified and categorised.

Designing a mobile app

Creating appropriate documentation is an important and necessary part of the mobile app design process. If you are developing individually, it helps organise your thoughts and planning. If working as part of a team, it helps to communicate ideas, share problem solving and identify potential problems as they emerge.

Design documentation should minimally contain:

- actual user requirements
- a proposed solution.

User requirements

User requirements define the problem you are trying to solve. In fact, no attempt at problem solving should be made before you fully understand what it is that the user actually wants. Sometimes, it is necessary to narrow down user requirements based on available resources or simply confirm your understanding of them more comprehensively, usually by asking additional questions of the client or through market research of potential users.

Proposed solution

The proposed solution represents a blueprint of how the mobile app is going to be built. The design documentation should include all the details of the blueprint. In order to create a comprehensive software blueprint, there are many elements that you must include. These elements are outlined in the sections below.

Description of program tasks

The description of program tasks is a list of the core functionality of the mobile app, generated from the actual user requirements.

The tasks performed need not be listed in a chronological order; this is often impossible to achieve as functionality on a mobile application may be accessed in different ways. However, it should be comprehensive, meet the user's requirements as fully as possible and be usable as a 'to do' checklist once formal development begins.

Target platform(s)

The platform should identify the:

- required mobile device(s)
- operating system(s) including the targeted version
- type of app (native, web or hybrid).

Mobile devices include various smartphones, tablets or wearable technology but you may also need to specify particular versions. For example, if your app needs a front and rear-facing camera to work properly, this may limit it to certain models. Some apps may be designed for particular screen sizes; if this is true, make sure that the design documentation makes this clear.

In addition, although your mobile app development will certainly target a specific operating system (e.g. Android™ rather than Apple® iOS), you must ensure that any key software feature that you have used is not operating system version specific (e.g. Android™ 6.1) unless it is unavoidable. Doing so will limit potential users or force them to upgrade their operating system, if this is possible; some devices will simply become too outdated.

Screen layouts and navigation

Visual design is the cornerstone of good mobile app development. Visual design includes the principles of good screen layout and intuitive user navigation.

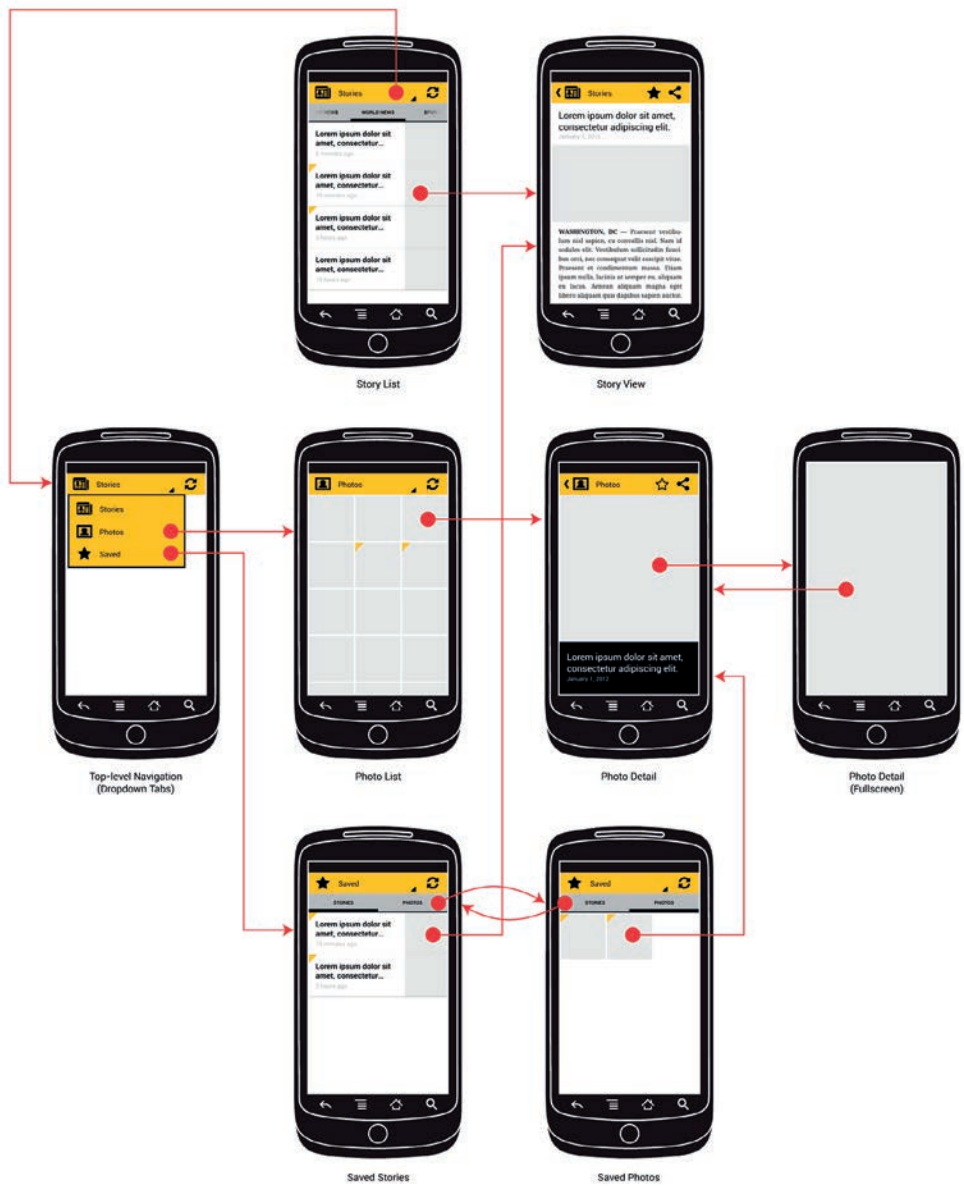


Figure 17.8: Android™ wireframe examples

Many designers use graphical mock-ups of devices, along with their screens and widgets to prototype an app and receive feedback before any lines of program code are actually written. There exist online design suites that permit developers to prototype their screen layouts and navigation using simple 'drag and drop' functionality. The industry term for this process is **wire-framing** (see Figure 17.8).

Key term

Wire-framing – an important step in the screen design process, helping the developer to plan layout and user navigation using paper-based or electronic models of the devices and their visual components.

Research

Visit a sample wire-framing tool to explore the rapid creation of visual prototypes for your mobile app. Balsamiq® is one such tool and has a free demonstration available online. To access this website go to: https://balsamiq.com/

Algorithms

An algorithm is simply a set of instructions that can be followed to solve a problem or perform a calculation. Apps may be made from many different algorithms, implemented in the program code using a combination of many different programming constructs, functions and procedures. Algorithms used in mobile apps may be represented using a number of different design tools such as:

- pseudocode – an informal English-like outline of a program which can be converted to the target programming language
- activity charts – also known as unified modelling language (UML) activity diagrams, these demonstrate user activity flow through a coordinated set of actions, for example using a standard notation to login, purchase an item or book an appointment
- flowcharts – a graphical representation of the program, showing its actions and logic through a set of standardised symbols.

These design tools are not specific to mobile app development. The skills that you build up using them can easily be transferred to other programming environments.

Link

For more on different programming environments, see *Unit 14: Computer Games Development* and *Unit 15: Website Development*.

Control structures

The algorithms that control mobile apps are typically built using a combination of three basic programming building blocks or control structures.

These control structures are sequence, selection and iteration.

1 Sequence – one action after another, none missed, none repeated (see Figure 17.9).

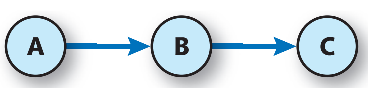


Figure 17.9: A sequence

2 Selection – actions that are chosen based on a supplied condition which is evaluated as true or false (see Figure 17.10).

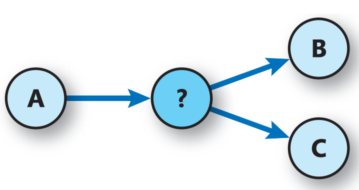


Figure 17.10: A selection

3 Iteration – actions repeated a number of times, typically, until a condition is no longer true (see Figures 7.11 and 7.12).

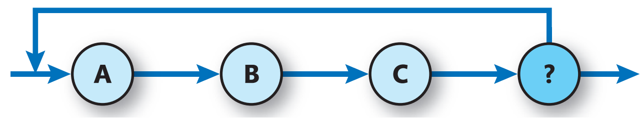


Figure 17.11: Post-conditioned iteration (or loop)

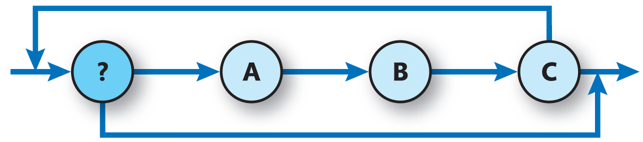


Figure 17.12: Pre-conditioned iteration (or loop)

Data validation

Mobile apps typically use widgets such as on/off switches, pickers and list boxes for input, removing (as much as possible) the need for traditional keyboard input.

When text-based input does occur, it usually needs to be validated. Validation is simply checking to see if something makes sense before it is processed: for example, 'Age' must be entered using numeric digits only.

An inputted 'Age' of 21 is acceptable, but a value of F4 or typing 'twenty one' is not. Building validation rules, which check whether different inputs are sensible, into a program is very important as failing to do so can cause inaccurate results or, more severely, a run-time error or a fatal application crash.

Integration of device capabilities

If specific device capabilities are required (functions, interface aspects, operating system features or certain permissions) they should be documented as part of the design. It is important to know how, when and where they will be used in your app.

Alternative solutions

Few solutions exist without alternative approaches that also have some merit. A critical part of the design document should be coverage of different design solutions and delivery plans for the app. These may approach the solution from the opposite direction, target different devices or potentially require more or less resources (time, money, expertise) to complete.

Having alternative solutions should see you exhausting possibilities and, crucially, having a contingency plan in place should events not go according to plan during development of your chosen solution.

Resources and assets

Your design should detail any existing resources and assets that need to be incorporated into your app, such as predefined code (yours and/or from a third-party library) and media assets.

Your media assets may include:

- images (e.g. .jpg or .png image files)
- video (e.g. .avi, .mpg, .mov video clips)
- audio (e.g. .wav, .mp3, .au, .aiff audo files).

Care should be taken to ensure that media assets are suitably processed before inclusion. For example, they should be cropped to appropriate size and **optimised** for efficiency. Listing them as part of the design documentation also acts as your checklist for content preparation.

Test and review schedule

Scheduling robust testing and reviews is a critical part of the design process. As we will see, the planning of thorough test plans and the selection of suitable test data is essential for ensuring that the app you have built performs reliably and as expected on the targeted device(s).

Review is best achieved through the analysis of selected user feedback. The aim of reviewing your mobile app design is to help improve and refine it before it is developed and then formally released. By reviewing the design, you should be able to iron out any defects or niggles before development.

Constraints

Constraints are limiting factors that are encountered on a personal or team level or imposed by your targeted platform.

For example, you may be constrained by time, development costs, the available technology, or simply by your own technical expertise or that of your team.

You may also discover that your design is constrained by the selected platform. The permissions, capabilities and limitations of the operating system or hardware may force you to make particular design decisions.

Legal and ethical considerations

Legal and ethical considerations must be taken into account as part of your design documentation, particularly those relevant in the United Kingdom. The Data Protection Act 1998 (DPA) in particular should determine the way in which your application handles personal data. Privacy is important to users, so you must think about how your application collects, deals with, controls and secures personal data, and if it is absolutely necessary to collect specific types of data in the first place.

Ethically, there are concerns over data that can be shared by companies who develop apps. The personal data that is collected could be used to influence insurance, credit, education or employment decisions. As noted previously, modern mobile devices generally rely on permissions to prevent the users' personal data from being used by the companies who own the apps without their consent but, ultimately, the developer's guiding ethics are the final protection.

In addition, pay attention to potential issues such as copyright infringement, especially with regard to the use of media assets such as images, video and audio files. Although media assets can be easily incorporated into your mobile app, they retain their original creator's copyright and remain their intellectual property (IP), not yours.

Key term

Optimised – optimised assets are created using file formats which are more efficient as they require less storage space. This can result in improvements in performance and a smaller digital footprint on the device's resources. Examples include using .jpg images rather than .bmp files as these use data compression to reduce file size.

Reflect

A local employer asks for a Health & Safety application that enables its employees to sign 'in' and 'out' of the building as they arrive and depart.

How could it be designed and what types of constraint affect the development of such an application?

Research

Visit the Information Commissioner's Office (ICO) for further reading on recommended codes of practice for mobile app developers, including ensuring users' privacy.

To access this website go to: **ico.org.uk**.

Visit the UK's legislative website and learn about the Copyright, Designs and Patents Act 1988. To access this website go to: **www.legislation.gov.uk**.

PAUSE POINT Can you explain what the learning aim was about? What elements did you find easiest?

Hint How do you define the design requirements for an app?

Extend Which elements should be included in the documentation of a proposed solution?

C Develop a mobile app that utilises device functions

Once the design of a mobile app has been reviewed against the user requirements and accepted by the client, it is possible to start the physical development process. There are a number of different phases to work through that will challenge you to develop a rewarding range of practical and technical skills in mobile app development.

Content preparation for mobile apps

Mobile apps are assembled using assorted forms of digital content, typically including program code, visual layouts, sound files, images (icons, pictures, animation) and video.

Before you build your mobile app, it is a good idea to prepare this content, that is, your resources, so that you have them ready to put straight into your app.

Step by step: Content preparation for mobile apps

5 Steps

1 Select appropriate applications and techniques to prepare your resources (see Figure 17.13).

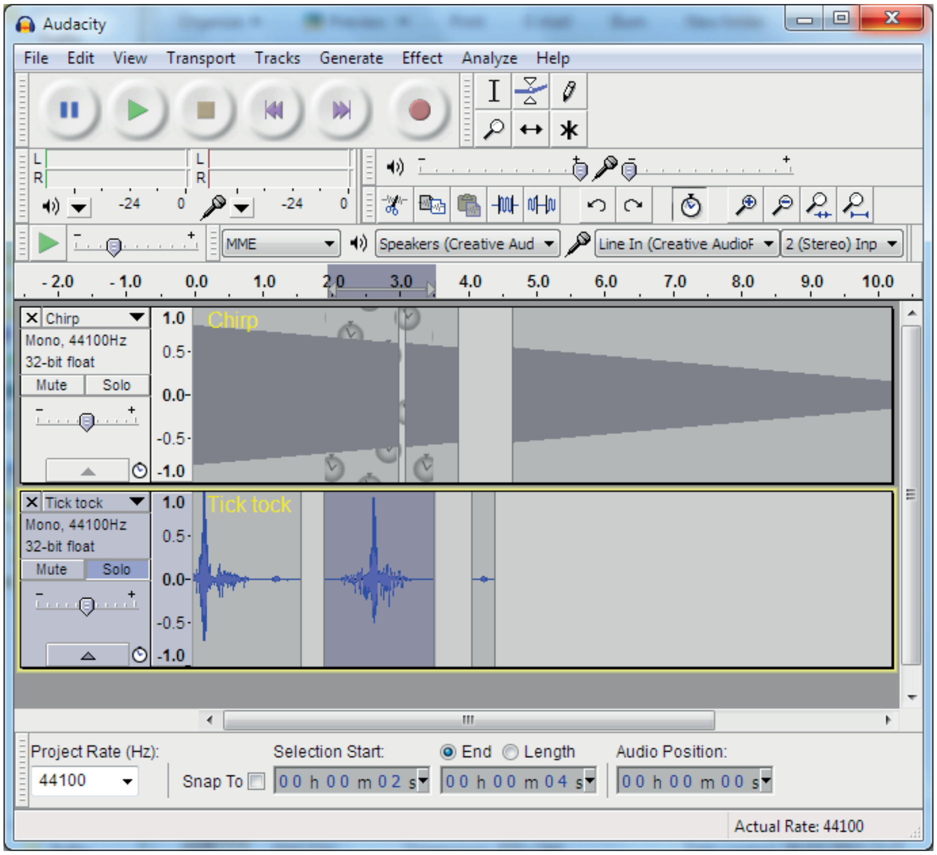


Figure 17.13: Audacity® is a popular, open source, audio recorder and editing suite

Your code development options are likely to be limited to either Android™ Studio or Apple® Xcode. However, there are many different options for editing images and audio files, including online utilities, freeware and commercial software. Most editing processes have suggested workflows that encourage best practice in order to achieve the best results for the specific devices you are targeting with your app.

2 Consider how different device attributes will affect your content when it is used (see Figure 17.14).

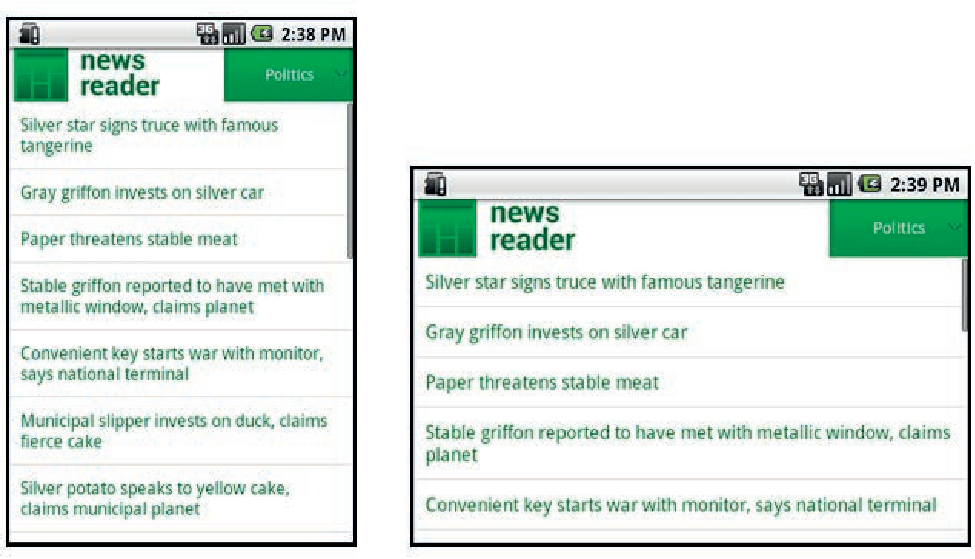


Figure 17.14: Same app, different device sizes and orientation

Things to consider include:

- orientation (landscape or portrait or both)
- physical screen size (e.g. 7 inches)
- screen resolution (in pixels) (e.g. 1024 × 600)
- available app resources (e.g. RAM (e.g. 1 Gigabyte (GB))
- sound (e.g. whether a device has Dolby® support).

3 Choose compatible file formats for media assets (see Figure 17.15).

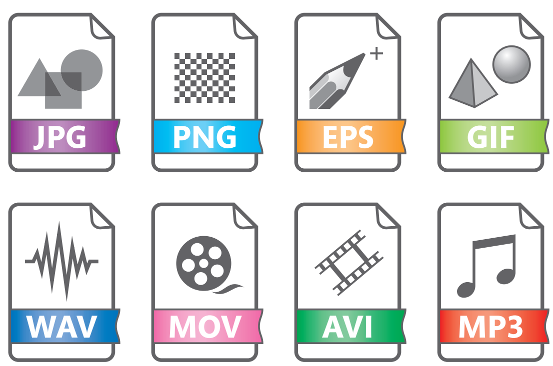


Figure 17.15: Different file formats – not all may be supported by your target device

App development (and certain programming languages) may only support files in a particular file format (e.g. images may only be .jpg or .png). Be sure that your media assets are available in the correct format. If they are not, you can usually convert them using specialist software or using online tools. You should also consider the effects of user interaction on assets. For example, a user zooming-in on a raster image will experience the image appearing 'blocky' or overly pixelated, whereas vector images (if supported) do not suffer from this issue.

4 Optimise content as appropriate (see Figure 17.16).

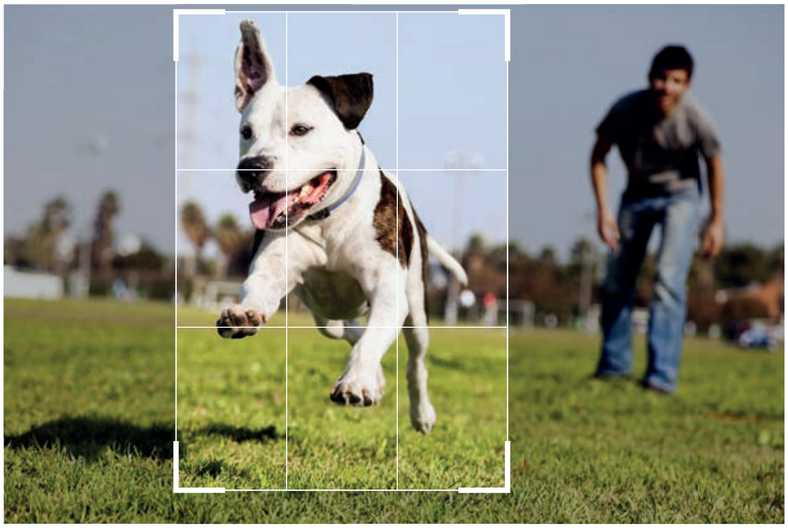


Figure 17.16: Optimising an image by cropping

Apply sensible optimisation to:

- program code (e.g. removing unnecessary sections of pre-written ('boilerplate') code that IDEs often provide as a helpful starting point
- reducing file sizes of media assets by using compression techniques such as saving images as .png or .jpg files, selecting the most efficient format for a particular type of content or cropping parts of files that are not needed.

You should always remember that there is a trade-off between optimisation levels and quality (be too aggressive and the quality of your content will suffer).

5 Think about security.

Some data or assets may contain sensitive information. Consider using encryption tools to protect them and the target users of your app from cyber attack. Encryption is typically an arithmetic algorithm that scrambles sensitive data using a user-defined key. If the data is encrypted, it is considered to be safe from third-party prying because it cannot be correctly unencrypted without the user's key.

Reflect

Designing any mobile app involves substantial planning, particularly in terms of investigating and understanding the target user's requirements and the complexities of the target platforms. Any proposed solution should not just list the actions you need it to perform but should include an estimate of the amount of time each step requires for development.

Recording your findings accurately and comprehensively throughout the design and content preparation stages strengthens your problem solving by providing a solid foundation on which effective development may begin.

Tip

Always keep your user keys (passwords) safe and change them regularly. Even encrypted data could be at risk if passwords are not secure.

Link

For more on graphics (raster and vector images) see section Developing Computer Games (Worked Example) in *Unit 14: Computer Games Development*.

Developing a mobile app

This section will take you through the process of developing a mobile app to meet identified user requirements and, as part of that journey, introduce you to the key concepts that form a mobile programming language and the development environment used to build it. This example uses Android™ Studio and the JavaScript programming language.

Reflect

Good communication skills are necessary for investigating and recording the target user's requirements. This may involve written and verbal communication, evidenced through activities such as conducting personal interviews, sending and responding to emails and the creation of formal design documentation for the mobile app.

Target users often know what they want in terms of a product but have very little technical awareness of the mobile app development process. You will have to reflect and learn how to adjust your verbal delivery to avoid the use of unnecessary jargon and find the language and technical level most suitable for the intended audience.

The following example will show you how to design a simple temperature conversion application that will allow the user to convert between readings in degrees Celsius and degrees Fahrenheit. The basic design can be seen in Figure 17.17.

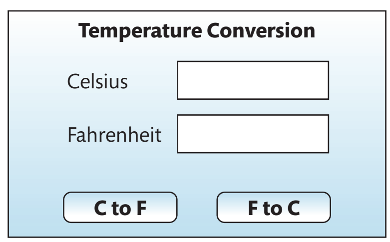


Figure 17.17: A simple design for a temperature conversion app

In order to create this app you will need to download and install the JavaScript SDK and a copy of Android™ Studio, both of which are freely available. The JavaScript SDK should be installed first, followed by Android™ Studio.

Link

Download JavaScript SDK from www.oracle.com

Once you start Android™ Studio and create a new project, you are usually presented by a new project window (on the left). If not, select the Project vertical tab and then expand 'res' and then 'layout'.

Clicking on the content_main.xml entry should display your empty app with the default 'Hello World!' TextView in the central pane, as shown in Figure 17.18.

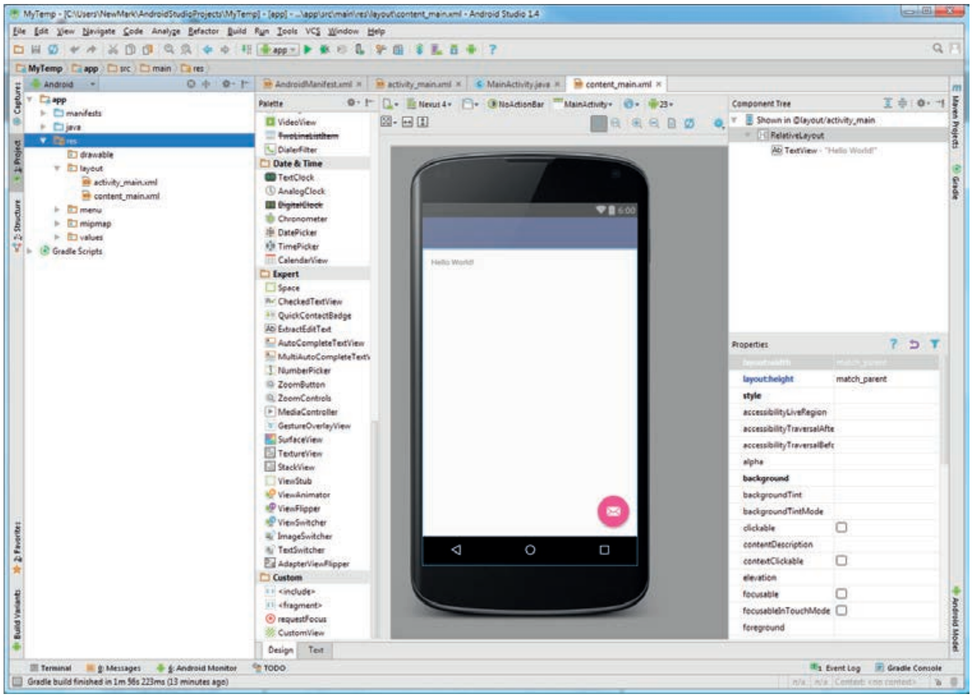


Figure 17.18: Android™ Studio design view, showing the default 'Hello World!' TextView (Google and the Google logo are registered trademarks of Google Inc., used with permission)

Android™ Studio's design view works on a simple 'drag and drop' principle, so you should be able to move the selected TextView freely around on the app's main form.

Android™ Studio has a number of different classes that may be used to place objects on the app's main form. These are shown in the Palette to the left of the central design pane. Classes are grouped into the following types.

- Layouts – control how the contents of a form are organised (e.g. in a table, in a grid, in rows).
- Widgets – different types of form element that are used to build the app's interface, allowing the user to input data, make selections and see output (e.g. TextViews, Buttons, CheckBox).
- Text Fields – specific types of TextView for defined jobs (e.g. Password, E-mail address, Telephone number).
- Containers – ways of grouping form elements together.
- Date & Time – different types of form elements related to the device's calendar and clock (e.g. DataPicker, TimePicker, CalendarView, TextClock, AnalogClock).
- Expert – complex types of form element for the more advanced app developer.
- Custom – a specialised class created by the developer or a third-party widget.

When you drag and drop a class from the Palette onto the form, JavaScript creates a solid object (a concrete instance of that class) which has a name, properties (things that describe it) and methods (things that it can do). Android™ Studio is also generating an XML (eXtensible Markup Language) file which describes the app's appearance as you make each change and addition; this is viewable on the Text tab at the bottom of the screen.

You will now delete the 'Hello World!' TextView and replace it with content based on your temperature conversion app design. This is shown in Figure 17.19.

- Edit the form's properties so that it resembles the design shown in Figures 17.17 and 17.19.

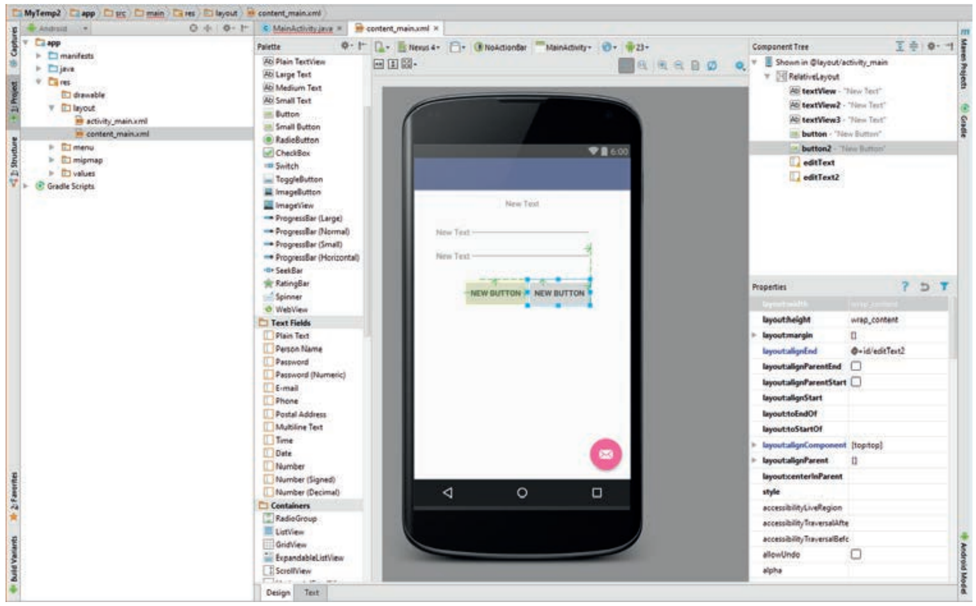


Figure 17.19: The main form with additional JavaScript objects dragged from Android™ Studio's Palette
(Google and the Google logo are registered trademarks of Google Inc., used with permission)

Each object can be edited by changing its properties (notice the Properties window in the bottom-right hand corner of the Android™ Studio IDE). Properties controls each object's visual appearance and allows you to alter their size, typeface and colour. It is also possible to double-click on a form object to edit its text and ID.

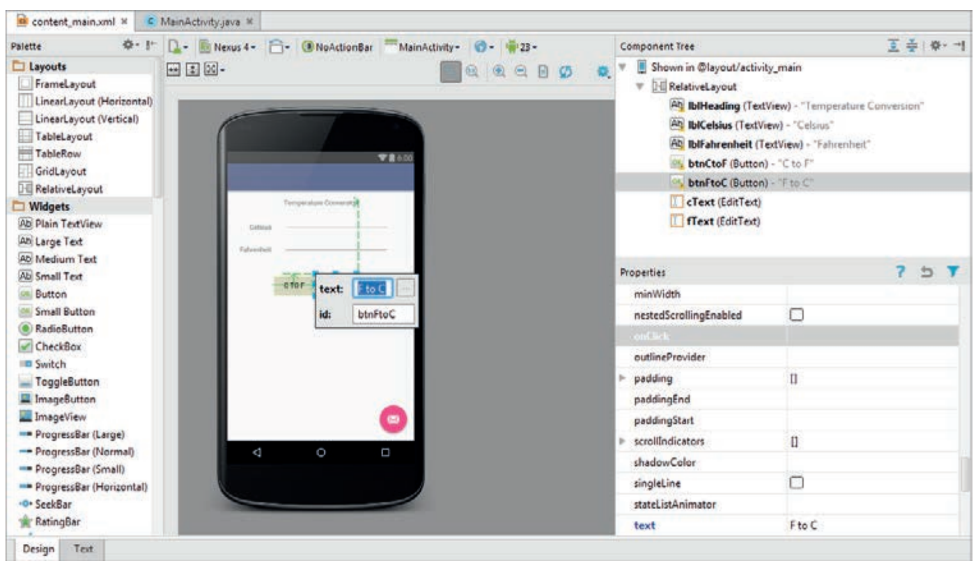


Figure 17.20: Android™ Studio design view, showing form objects with new properties and IDs
(Google and the Google logo are registered trademarks of Google Inc., used with permission)

You may have noticed that each object on your app's main form has a default name (or ID), for example textView, textView2, editText, editText2. Although this is helpful, it does not generate readable program code so it is a good idea to rename them sensibly. Do this by double-clicking on each object; a small dialog will pop up allowing you to change their IDs, as shown in Figure 17.20.

Make sure the new IDs match the list of objects, as shown in the Component Tree window in Figure 17.20.

Programming constructs

Now pause and focus on the JavaScript programming language in which Android™ apps are written.

Reserved words, local and global variables, constants and assignment

JavaScript, like most programming languages, has a number of reserved words. Reserved words are fundamental parts of the language which *cannot* be used by you for naming things. A list of common JavaScript reserved words is found in Table 17.4.

Table 17.4: Alphabetical list of JavaScript's reserved words (some are version specific)

`abstract`	`continue`	`for`	`new`	`switch`
`assert`	`default`	`goto`	`package`	`synchronized`
`boolean`	`do`	`if`	`private`	`this`
`break`	`double`	`implements`	`protected`	`throw`
`byte`	`else`	`import`	`public`	`throws`
`case`	`enum`	`instanceof`	`return`	`transient`
`catch`	`extends`	`int`	`short`	`try`
`char`	`final`	`interface`	`static`	`void`
`class`	`finally`	`long`	`strictfp`	`volatile`
`const`	`float`	`native`	`super`	`while`

When you name things, you are creating identifiers. There are two types of basic identifier – variables and constants. As the names suggest, variables store values which your app can manipulate. Constants store values that cannot change while the app is running.

Discussion

Mobile apps often use many variables and constants. Imagine creating a friendly app to calculate your wages for a part-time job. In a small group, discuss which values would be stored in variables and which would be constants, making sure that you justify your decisions.

In order to declare a variable, we need to specify its name and its **data type**. Developers must select the most appropriate type for the data they wish to store. Table 17.5 shows JavaScript's primitive data types and examples of different declarations.

Key term

Data type – the kind of data that we want to store in the mobile device's RAM. Storing each type of data requires different quantities of RAM. For example, the data type needed to store a number is different from the one used to store a character and will require different amounts of RAM. Exact names of data types can vary between different programming languages, so be careful to select the right one.

In each declaration we are saying:

data type variable name = value

The = sign (or assignment operator; see Table 17.5) is used to store the value in the named variable. The first time you do this is called an 'initialisation'; afterwards it is simply called 'assignment'. Variables may change their value many times as your app runs.

Link

For more on RAM see Memory and storage in *Unit 14: Computer Games Development*.

Table 17.5: JavaScript's primitive data types, their features and example declarations

Data type	What it is	Min – max range (where appropriate)	Default value	Example declaration
`byte`	8-bit signed two's complement integer	-128 to 127	0	`byte age = 17;`
`short`	16-bit signed two's complement integer	-32,768 to 32,767	0	`short qty = 10000;`
`int`	32-bit signed two's complement integer	- 2,147,483,648 to 2,147,483,647	0	`int largeqty = 100000;`
`long`	64-bit signed two's complement integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	0L	`long vlargeqty = 100000L;`
`float`	single-precision 32-bit IEEE 754 floating point		0.0f	`float wages = 2814.20f`
`double`	double-precision 64-bit IEEE 754 floating point		0.0d	`float salary = 28140.20d`
`boolean`	Boolean 1-bit state; true or false	true or false	false	`boolean alive = true;`
`char`	single 16-bit Unicode character	'\u0000' to '\uffff'		`char initial ='A';`

Most programming languages have the concept of local and global variables. The easiest way to think about this is to remember that global variables can be used anywhere in your app's code. Local variables are limited to being used in the block of code in which they are declared. However, JavaScript does things a little differently as you will see.

Key term

Operator – a special symbol (or multiple symbols) which tells the program to perform specific arithmetic, relational or logical operations on its data. Operators must be used in a specific order of precedence (this describes which is executed first). You may be familiar with this concept from using BIDMAS (or BODMAS) (Brackets, Indices/Orders, Divide, Multiply, Add and Subtract) in mathematics.

Operators

Operators are special symbols used to perform special tasks in most apps. If you are already familiar with symbols such as using * for multiply, then you are familiar with an arithmetic operator. JavaScript has a number of these and it is a good idea to familiarise yourself with them. The most commonly used operators in JavaScript are shown in Table 17.6.

Table 17.6 : Common JavaScript operators, showing their grouping and purpose

Java® operators	Name	Operator group	Purpose
+	Add	Arithmetic	Adds two values giving their sum
-	Subtract	Arithmetic	Subtracts one value from another giving their difference
*	Multiply	Arithmetic	Multiplies two values giving their product
/	Divide	Arithmetic	Divides one value by another giving their quotient
++	Increment	Arithmetic	Increases a value by 1
--	Decrement	Arithmetic	Decreases a value by 1
==	Equal to	Relational	Tests where two values are equal
!=	Not equal to	Relational	Tests whether two values are unequal
>	Greater than	Relational	Tests whether one value is greater than another
<	Less than	Relational	Tests whether one value is less than another
>=	Greater than or equal to	Relational	Tests whether one value is greater than or equal to another

Table 17.6: – *continued*

<=	Less than or equal to	Relational	Tests whether one value is less than or equal to another
=	Simple assignment	Assignment	Assigns a value to a variable, not to be confused with the double equal sign which tests equality
&&	And	Logical	Performs a logical And operation
‖	Or	Logical	Performs a logical Or operation
!	Not	Logical	Performs a logical Not operation

Research

It is not possible to list all of JavaScript operators here. Visit the following online resource to find out about the other operators that exist and gather examples of their usage:

www.tutorialspoint.com/java/java_basic_operators.htm

Control sequences

Recall the three control structures introduced earlier: sequence, selection and iteration.

When you are building a mobile app, sequences are represented by any block of JavaScript program code that executes line-after-line, with none missed or repeating.

Research

You can quickly experiment with the code samples shown by visiting a free, online JavaScript compiler. When you have keyed in the program code you want to try, simply click on the "Compile" and "Execute" options. You may even save your code for future use.

Visit **www.tutorialspoint.com** to try this out and discover new coding techniques, for example nested if statements and conditional operators.

Link

For more on sequence, selection and iteration see Control structures.

Selections are built using either an if...else or a switch statement. Which one you use depends on what you are trying to achieve.

A simple if...else statement has a condition that evaluates to either true or false. As shown in Figure 17.21, the actions in the first block are executed if the condition is true, otherwise the actions in the (optional) else block are used. The output from two test runs of this JavaScript code is shown in Figure 17.22.

```
import java.util.Scanner;

public class NumberTest {

    public static void main(String []args) {

        int number1;
        int number2;

        Scanner keyboard = new Scanner(System.in);
        System.out.print("Enter 1st number: ");
        number1 = keyboard.nextInt();
        System.out.print("Enter 2nd number: ");
        number2 = keyboard.nextInt();

        if (number1 == number2) {
            System.out.println("Numbers are the same!");
        }
        else {
            System.out.println("Numbers are NOT the same!");
        }
    }
}
```

Figure 17.21: JavaScript's if...else statement

```
Enter 1st number: 4
Enter 2nd number: 5
Numbers are NOT the same!

Enter 1st number: 78
Enter 2nd number: 78
Numbers are the same!
```

Figure 17.22: Output from two different tests of JavaScript's if...else statement

Tip

Many modern programming languages are case sensitive and JavaScript is no exception. If you are having problems compiling your programs, check that you have used upper case and lower case letters correctly in your code. Missing out important symbols such as the semi-colons (which are used in JavaScript to separate statements) is also a common error, even for experienced developers.

JavaScript's switch statement is a useful way to check for multiple values at the same time, as shown in Figure 17.23. It is also possible to add a default check in case no listed options are matched. Output from four test runs of the switch statement is shown in Figure 17.24.

```
import java.util.Scanner;

public class BankAccount {

    public static void main(String []args) {
        int option;

        Scanner keyboard = new Scanner(System.in);
        System.out.println("1 - Bank Balance, 2 - Account Query, 3 - Payments");
        System.out.print("Enter option 1, 2 or 3: ");
        option = keyboard.nextInt();
        switch (option) {
            case 1: System.out.println("You have chosen to see your balance");
                    break;
            case 2: System.out.println("You have chosen to make a query");
                    break;
            case 3: System.out.println("You have chosen to make a payment");
                    break;
            default:System.out.println("You have not chosen a correct option!");

        }
    }
}
```

Figure 17.23: JavaScript's switch statement

```
1 - Bank Balance, 2 - Account Query, 3 - Payments
Enter option 1, 2 or 3: 1
You have chosen to see your balance

1 - Bank Balance, 2 - Account Query, 3 - Payments
Enter option 1, 2 or 3: 2
You have chosen to make a query

1 - Bank Balance, 2 - Account Query, 3 - Payments
Enter option 1, 2 or 3: 3
You have chosen to make a payment

1 - Bank Balance, 2 - Account Query, 3 - Payments
Enter option 1, 2 or 3: 8
You have not chosen a correct option!
```

Figure 17.24: Output from four different tests of JavaScript's switch statement

Iterations (sometimes called loops) are represented by one of three common statements (for, while and do...while). Figures 17.25 to 17.27 show JavaScript's for, while and do...while loops for a particular example. Figure 17.28 shows that the output from all three loops is identical.

```java
public class ForLoop {

    public static void main(String []args) {
        int counter = 0;

        for (counter = 1; counter <= 5; counter++) {
            System.out.println("BTEC National");
        }
    }
}
```

Figure 17.25: JavaScript's for loop

```java
public class WhileLoop {

    public static void main(String []args) {
        int counter = 1;

        while (counter <= 5) {
            System.out.println("BTEC National");
            counter++;
        }
    }
}
```

Figure 17.26: JavaScript's while loop

```java
public class DoWhileLoop {

    public static void main(String []args) {
        int counter = 1;

        do {
            System.out.println("BTEC National");
            counter++;
        } while (counter <= 5);
    }
}
```

Figure 17.27: JavaScript's do...while loop

```
BTEC National
BTEC National
BTEC National
BTEC National
BTEC National
```

Figure 17.28: Output from all three loops is identical

Although all three loops generate the same output, they have all worked differently. Ideally, you would use for loops when you know that code needs to repeat a fixed number of times. While loops may not even run once (they are pre-conditioned) and do...while loops always repeat at least once (they are post-conditioned).

Loops repeat based on their controlling condition being true. In the example shown in Figures 7.25 to 7.28, the controlling condition is the counter variable having a value less than (or equal to) 5. When the counter reaches 6, the condition is no longer true and the loop ends.

Functions and procedures

Another key component of a mobile app is the concept of a function or procedure. **Functions** and **procedures** are common features in procedural programming while, in **object-oriented** programming, they are essentially represented by the methods which occur with the class.

Key terms

Function or procedure – a function is a block of code, ideally between 5 and 50 lines in length, which has a single defined purpose. Although written just once, it can be executed many times during a program by using a function 'call', reducing the need for repeated code. In some programming languages, the terms 'function' and 'procedure' are used interchangeably, but, in others they are very different concepts: a function typically returns a calculated value while a procedure performs a single identifiable task.

Object-oriented – JavaScript, which we use to create Android™ apps, is categorised as a class-based, object-oriented programming (OOP) language. In OOP, objects are created from classes that are usually modelled on real-world 'things'. Each class acts as a software blueprint, encapsulating (or containing) the thing's state (its data or properties) and behaviour (its functions or methods) in program code.

Tip

The temperature conversion app is relatively straightforward and should just make use of a basic sequence control structure. When you start designing more complex apps, the other constructs (sequence and iteration) will become important, so remember the if, switch, for and while statements shown here.

Objects and classes

Classes are a fundamental aspect of every Android™ mobile app solution created using JavaScript. Classes are used to represent the physical components that you use to build an app. For example, each item in the Android™ Studio Palette (screen components such as forms, Layouts, Widgets, and Text Fields) represents a different class that you can use.

Once the class is dragged to the form, a concrete instance of that class is created, which we call an object. Try to think of a class as a design pattern, something like a jelly mould, stencil or cookie cutter; one jelly mould (the class) can be used to make many jellies (the objects).

Each object has associated methods and properties and can interact with other objects to perform a task. Commercial apps will use a wide range of classes, many of which may be freshly created by their developers.

Event handling

Once you have created the visual design for your app's interface and named objects sensibly, the next step is to code the algorithms that calculate the outputs or actions needed. In the case of our temperature conversion app, this means converting between a temperature input in degrees Fahrenheit to an output in degrees Celsius and vice versa.

These calculations are executed when a user presses one of the two buttons we have added to the app's main form. When the user presses a button (or performs any action the operating system recognises) it triggers a specific event – in this case, what is commonly called an 'on click' event.

In order to link the event trigger to the calculations, you need to build an event handler. There are many ways to achieve this in Android™ Studio. The most traditional method involves registering a listener method for each button that will run a specifically coded **event handling** method.

Key term

Event handling – describes the process of using a specially written function or class method to perform set actions when a user or system event is triggered. For example, outputting the result of a calculation when a button is pressed or displaying a device's 'low battery' warning indicator when it falls below a certain charge level.

Revisit the Project explorer pane, expanding 'Java' and then 'MainActivity'. Clicking on this should display the JavaScript program code, which will have been automatically created for you. You will need to insert the highlighted code shown in Figure 17.29 to add appropriate event handlers that will respond to user inputs and calculate the desired outputs, in order to get the right screen components.

```
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener((view) → {
                Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
        });

        addButtonCtoFListener();     //call method to add listener to CtoF button
        addButtonFtoCListener();     //call method to add listener to FtoC button
    }
```

Figure 17.29: Adding new code to your app

The new (highlighted) lines shown in Figure 17.29 perform two actions.

1 They import various JavaScript code packages that you need for your app to compile into your program.
2 They call two listener methods, one for each button on the app's form.

Your next task is to create the JavaScript code for the two event listener methods. Each will include an event handler that:

- gets the value entered into a TextField on the app's form
- performs the correct calculation
- puts the results back into the other TextField on the app's form.

Figures 7.30 and 7.31 show the new code for each listener and handler. You should note that the JavaScript code for the addButtonCtoFlistener method is annotated to demonstrate best practice.

```
/*
 * Class method adding a Listener to
 * the CtoF button to enable responding to
 * events
 */
public void addButtonCtoFListener() {

    //Create button object representing CtoF button on form
    Button btnCtoF = (Button)findViewById(R.id.btnCtoF);

    /*
     * OnClick listener for CtoF button will
     * call its associated event handler to perform
     * the CtoF calculation
     */
    btnCtoF.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            float celsius;       // Inputted Celsius value
            float fahrenheit;    // Calculated Fahrenheit value

            //alias Celsius input box
            EditText cValue = (EditText) findViewById(R.id.cText);

            //grab celsius value from input box, converting from text to floating point
            celsius = Float.valueOf(cValue.getText().toString());

            //perform conversion of Celsius to Fahrenheit
            fahrenheit = (celsius * 9.0f / 5.0f) + 32.0f;

            //alias Fahrenheit input box
            EditText fValue = (EditText) findViewById(R.id.fText);

            //convert calculated Fahrenheit floating point value to text
            String fahrenheitText = Float.toString(fahrenheit);

            //put the converted Fahrenheit text value into the Fahreneit input box
            fValue.setText(fahrenheitText);
        }
    });
}
```

Figure 17.30: JavaScript code for the addButtonCtoFListener method (annotated)

```
public void addButtonFtoCListener() {

    Button btnFtoC = (Button)findViewById(R.id.btnFtoC);

    btnFtoC.setOnClickListener(new OnClickListener() {
        public void onClick(View v) {
            float celsius;
            float fahrenheit;
            EditText fValue = (EditText) findViewById(R.id.fText);
            fahrenheit = Float.valueOf(fValue.getText().toString());
            celsius = (fahrenheit - 32.0f) * 5.0f / 9.0f;
            EditText cValue = (EditText) findViewById(R.id.cText);
            String celsiusText = Float.toString(celsius);
            cValue.setText(celsiusText);
        }
    });
}
```

Figure 17.31: JavaScript code for the addButtonFtoCListener method (not annotated)

Reflect

Read through the annotated code for the addButtonCtoFListener method and compare it with the similar, but uncommented, code in addButtonFtoCListener. Could you add appropriate comments to the addButtonFtoCListener method to improve its readability?

Code annotation

As you have seen, it is important to ensure that your code is suitably documented. Annotating code essentially means inserting developer-readable comments throughout your code. Although these comments are removed during the compilation process and therefore inaccessible to the user, their inclusion is good practice.

The purpose of annotating your program code with meaningful comments is to improve its readability and aid understanding. This can be especially important when programmers are asked to redevelop apps that were originally written by others.

Your comments should make clear how your code relates to its real-world application, not explain the syntax of the actual reserved words used. The actual techniques used to comment on program code vary between program languages.

JavaScript uses a combination of multiline and single line comments, and this is shown in Figure 17.32. Another popular use of annotation is to 'comment out' program code that the developer wants to keep but not compile, possibly as they test new ideas or identify code which is not working correctly.

```
public class HelloWorld{

    /*
        Main function

        Program to test program output

        Written by A. Jones
        Version 1.0

     */
    public static void main(String []args){

        // Output welcome message
        System.out.println("Hello World");

    }
}
```

Figure 17.32: Multiline and single line comments

The use of meaningful identifiers is also a key aspect of code annotation as the representation names you have chosen are said to self-document.

Some SDKs and third-party tools can generate web page documentation automatically from your program code if it is formatted in the correct fashion. For example, Oracle®'s Javadoc tool performs this function for JavaScript.

Research

Visit **www.oracle.com** to investigate Oracle®'s Javadoc tool and learn how to incorporate Javadoc-compatible comments into your code and automatically generate navigable web pages for them.

Utilise device capabilities

Android™ and Apple® support mobile app developers by providing **application programming interfaces (APIs)** and **frameworks** that permit apps to receive motion data from integrated device hardware such as gyroscopes and accelerometers. Android™ calls this the Sensors framework while Apple® refers to their implementation as the Core Motion framework.

Key terms

Application programming interface (API) – this acts as a library of pre-written routines that provide the developer with access to other systems (such as databases), the operating system of a device or its actual hardware. The aim of an API is to make the development process easier by abstracting (hiding) the complexity of the software and hardware systems beneath.

Framework – a particular set of development tools that can be accessed via its API.

Interrogate device status

A key part of mobile app programming is the ability to interrogate the device's status. Different types of status may be checked by the developer, including important statistics such as its physical location, battery level or orientation (that is, whether it is portrait or landscape). For example, Android™ uses a useful BatteryManager class, which provides a method for querying battery and charging properties.

Research

Many examples can be found that provide the JavaScript code necessary to interrogate an Android™ device's status. Visit the **www.tutorialforandroid.com** website that demonstrates how the battery level of such a device may be checked.

Investigate this code, adapt it and incorporate into your own projects.

Although you do not need to perform this type of interrogation in your introductory temperature conversion app, being able to do so is an important skill.

Orientation of device

Your app needs to auto-detect the orientation of the physical device on which it is running (and when it changes). It does this by querying the screen's rotation. The device's normal rotation is derived from its screen format: for example, if the screen is naturally tall (portrait) and the device is turned sideways (landscape) its rotation value will either be 90 or 270 degrees depending on the direction it was turned (clockwise or anticlockwise). Helpfully, Android™ supplies an appropriate class called 'Display' within its API to assist developers to determine the screen rotation, size and refresh rate.

Apps can also force the orientation mode by automatically changing the device's rotation to fit a portrait or landscape view, although this is not generally considered to be good practice.

Creating an executable for a target device

You can deploy your app to either an emulated device or the actual physical device you have targeted.

Emulated device

Using an emulated device is useful because it lets you test the compatibility and performance of your app on types of device to which you may not have physical access, for example different smartphones, tablets, TVs and pieces of wearable technology. In Android™ Studio, you can achieve this by creating an **Android™ virtual device (AVD)**. To create an AVD it is necessary to specify the hardware, and the version of the Android™ operating system being used (4.X KitKat, 5.X Lollipop or 6.0 Marshmallow).

Key term

Android™ virtual device (AVD) – a software emulated version of a particular physical device, running within a host operating system such as Microsoft® Windows® operating system or Apple® OS X®.

Android™ Studio allows you to create multiple AVDs and then choose which one you want to use when developing your app. This allows you to test your app easily on many different physical devices. The device can be chosen through the available hardware definitions, as shown in Figure 17.33.

Tip

Emulator vs Real Device testing

- Emulation may be slower than the real device.
- Many different devices may be tested through emulation which may be otherwise unavailable.
- Emulation options may be limited depending on the development hardware.
- Emulation represents no risk to a real device.
- No need to enable development mode on a device in order to test an app.
- Only real devices offer the truest app performance and user experience.

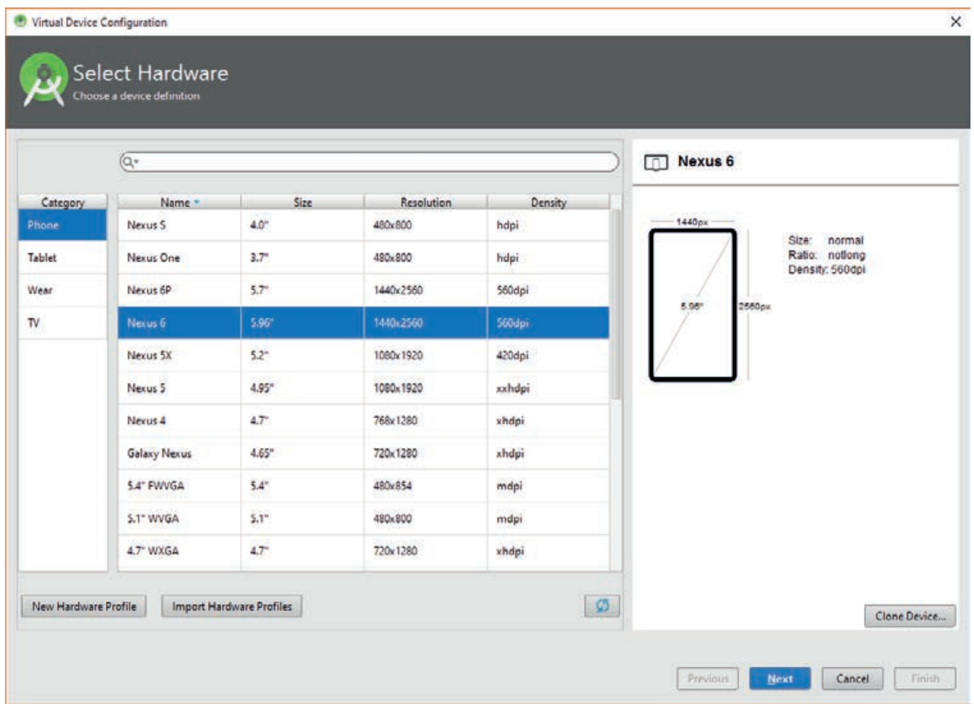
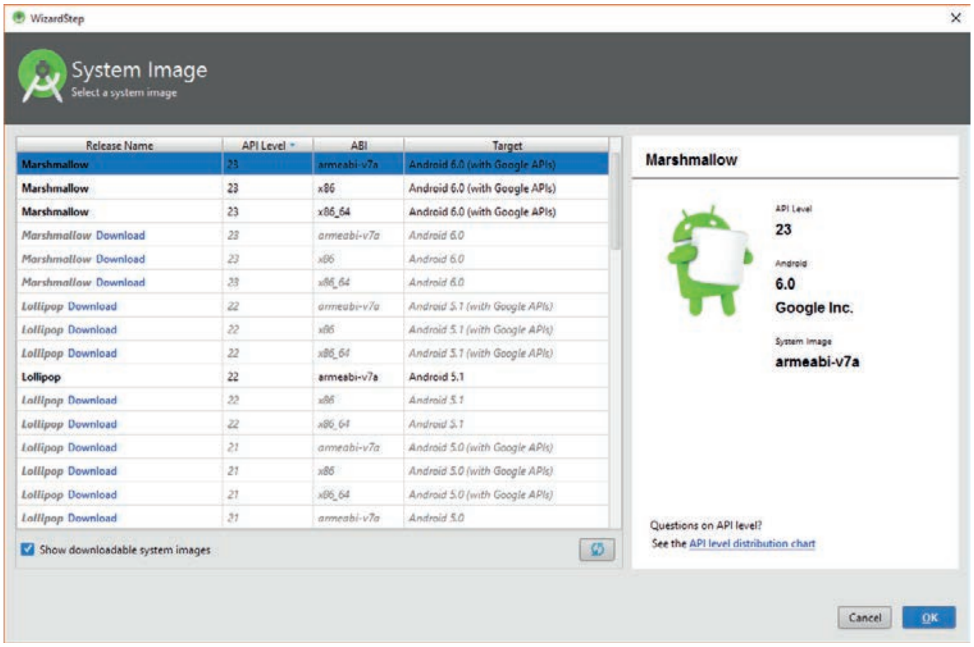


Figure 17.33: Selecting a hardware definition for a real Android™ device
(Google and the Google logo are registered trademarks of Google Inc., used with permission)

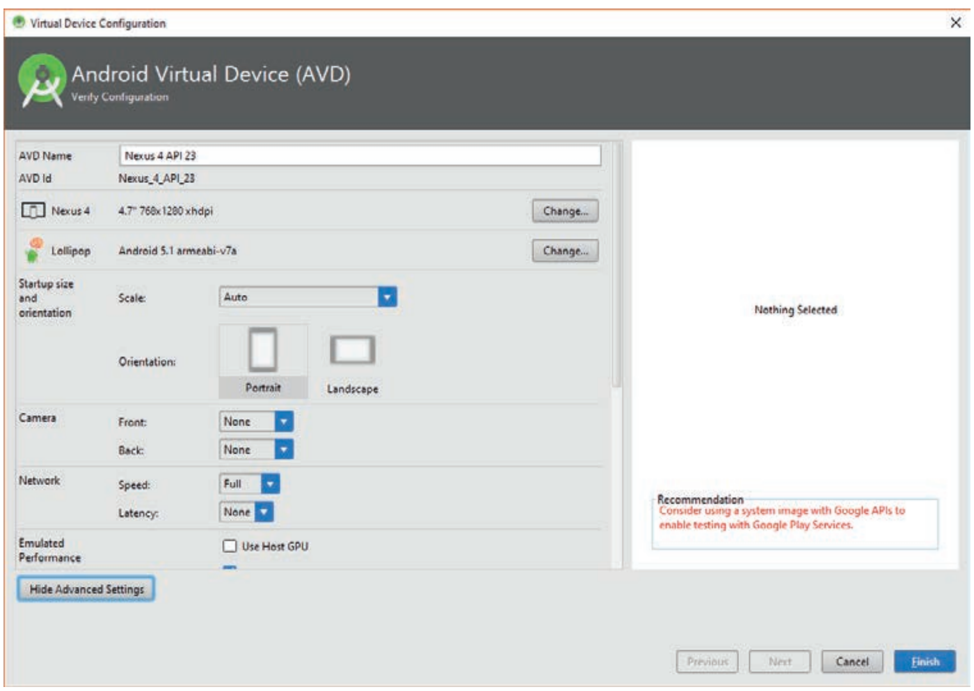
It is then possible to select an Android™ system image to use. Android™ Studio will have some images already installed but you may download others as you need them (as shown in Figure 17.34). In order to run your app successfully using an AVD, you must use a system image that is compatible with the programming or hardware features you have used.

Each system image is typically available in different virtualisation options: x86 (32-bit), x64 (64-bit) or ARM (Advanced Reduced Instruction Set Computing Machine). ARM is generally the slowest but is more likely to be independent of virtualisation options set in your host PC's BIOS (Basic Input Output System) (these should always be modified with caution).



▶ **Figure 17.34:** Selecting an Android™ system image for the AVD
(Google and the Google logo are registered trademarks of Google Inc., used with permission)

It is then possible to tailor various options to your personal preferences, as shown in Figure 17.35.



▶ **Figure 17.35:** Setting preferences for your AVD
(Google and the Google logo are registered trademarks of Google Inc., used with permission)

▶ **Figure 17.36:** An AVD running in Microsoft® Windows® 10 operating system

Figure 17.36 shows the AVD running in Microsoft® Windows® 10.

Physical device

The other technique for testing your Android™ app is to connect the system upon which you are developing it (for example Apple® Mac® or Microsoft® Windows® PC) to

▶ **Figure 17.37:** Android™ device connected via USB cable

a physical Android™ smartphone, tablet, or wearable technology via a compatible USB (Universal Serial Bus) cable, as shown in Figure 17.37.

In addition to installing the correct Android Debug Bridge (ADB) driver for your host system to talk to the Android™ device, it is often necessary to modify some of the device's settings in order to run your app. This often involves enabling options for USB or running apps not downloaded through Google Play™, and only has to be performed once. These developer settings can vary from device to device, and may even be deliberately hidden to prevent accidental damage, so you are advised to check the correct procedure with the device's manufacturer before attempting this.

The primary advantage of running your app on the target physical device is that you will get the truest impression of its performance and behaviour. Many devices also have developer debug tools that can assist your live testing.

As a developer, you just need to select the connected Android™ device (not an AVD) once you have clicked Android™ Studio's green 'run' app button. If the device is successfully connected, it should be listed as a running device, which can then be chosen. Once you have selected it and clicked the OK button, Android™ Studio will then start to transfer the necessary JavaScript code to the device via USB and start running the app; the process is relatively quick.

Figure 17.38 shows that a physical device has been selected as the target platform and Figure 17.39 shows the temperature conversion app transferred and running on a physical device.

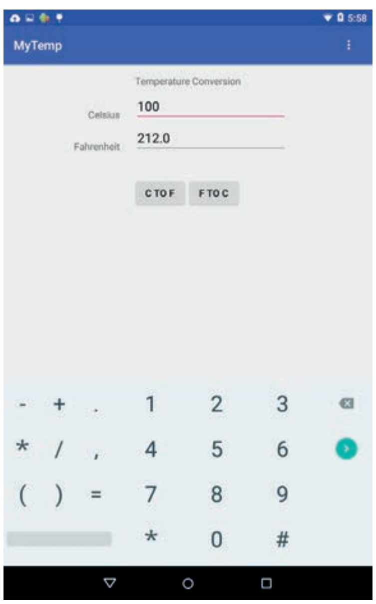


Figure 17.39: Your app transferred and running on a physical device
(Google and the Google logo are registered trademarks of Google Inc., used with permission)

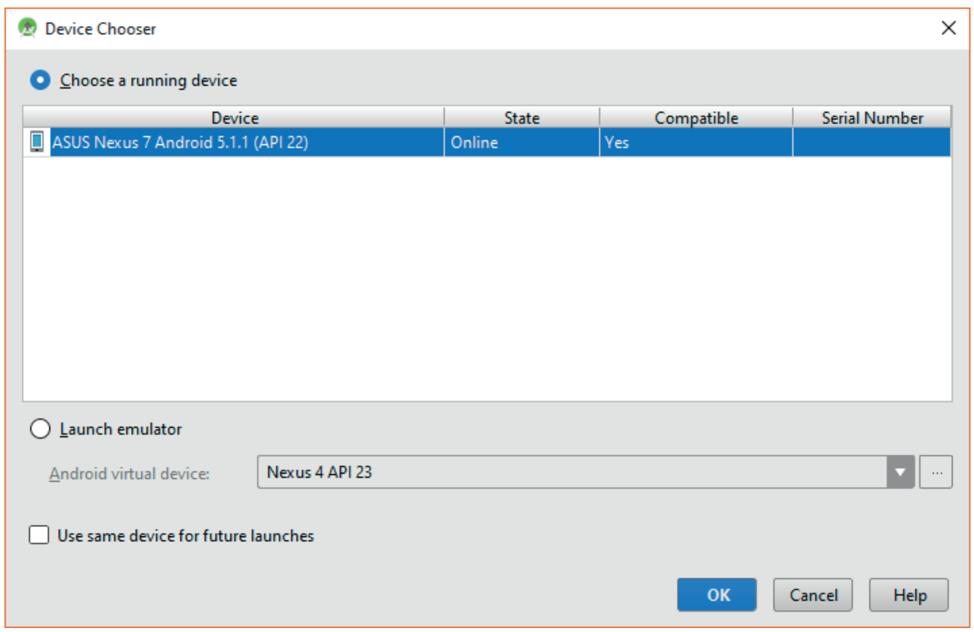


Figure 17.38: Selecting a physical device as the target platform

The physical device's virtual keyboard automatically appears as the app runs, limiting the user to entering only numbers. This happens because of the design decision to choose 'Number (Signed)' text fields.

Although a quick test seems to indicate that the app is working correctly (100 degrees Celsius is indeed equivalent to 212 degrees Fahrenheit), your next step should be to review your efforts through a process called quality control.

Quality control

Quality control is a process that reviews the standards of manufacturing applied during production. Its aim is to deliver the best product or service for a customer. For

a mobile app, this essentially means measuring how well the design was implemented during development for the target device(s).

Quality control is an intrinsic part of the mobile app development process, helping you to assess the resulting standard that your development process has achieved and, crucially, to identify areas of potential improvement. Quality control should be performed before formal testing starts. An overview of quality control is shown in Table 17.7.

Table 17.7: Quality control areas of concern, key questions and considerations

Area	Question	Consider...
Efficiency and performance (This is called application 'profiling')	How does your app impact on the resources of the mobile device? Can you identify any issues affecting performance?	• use of RAM allocation • use of device storage media • use of central processing unit (CPU), particularly in terms of its percentage share • use of system and application processes that may highlight performance bottlenecks • use of graphical processing unit (GPU), especially when rendering user interface (UI) elements • use of battery, comparing drain of similar applications • temperature constraints – mobile devices can shutdown to protect components during periods of heavy use (especially when recharging) if they become too hot.
Maintainability	How easy is it to modify or improve your app? Have you programmed the app appropriately?	• recommended standards of coding • levels and usefulness of code annotation • structure and organisation of your program code • extensibility of your solution • use of programming constructs, functions, procedures, data types, classes and the device's code framework.
Portability	Which devices are compatible with your app? How could you improve the potential compatibility?	• devices which will correctly run your app • devices which do not run your app (and identify the reason(s) why) • devices which run your app but have minor issues (and what can be adjusted to improve portability) • any unexpected compatibility issues which you had not anticipated • whether your app's actual portability meets the platforms targeted in the design phase (including specific versions of the operating system).
Usability	How easy is it for a user to interact with your application? How can you improve the user experience?	• adherence to manufacturer guidelines, ie Apple® iOS and Android™ recommended design principles and standards • capturing user feedback, both good and bad • identifying common likes and dislikes • requesting user feedback for improvement • identifying whether the app meets the user requirements targeted in the design phase.

Links

- For more on RAM, CPU and GPU see the section on Hardware in *Unit 14: Computer Games Development*.
- For a reminder of what is meant by 'user interface (UI)', see User interface.

Quality control is best achieved through manual (rather than automated) review and can be a self-reflective process. Formal testing will examine feedback from a wider range of users.

> **Reflect**
>
> The quality control process should help you to assess the standard of your app, both in terms of its design and its implementation. In doing so, it should provide a vital insight into your own performance levels, by highlighting the target areas that have been met and those which need some improvement.
>
> Reflective quality control is a key component of any self-improvement process and is a requirement for achieving higher grades and, eventually, of improved performance in the workplace. Reflect upon the design and implementation of the development of the Temperature Conversion App. How could it be improved?

Testing a mobile app

Testing is widely recognised as an essential process that ensures that any mobile app you develop meets the requirements that were identified during its design, and that it operates in an accurate, reliable and robust manner. Testing should be performed every time the app is updated, so the process should be seen as cyclical; this is shown in Figure 17.40.

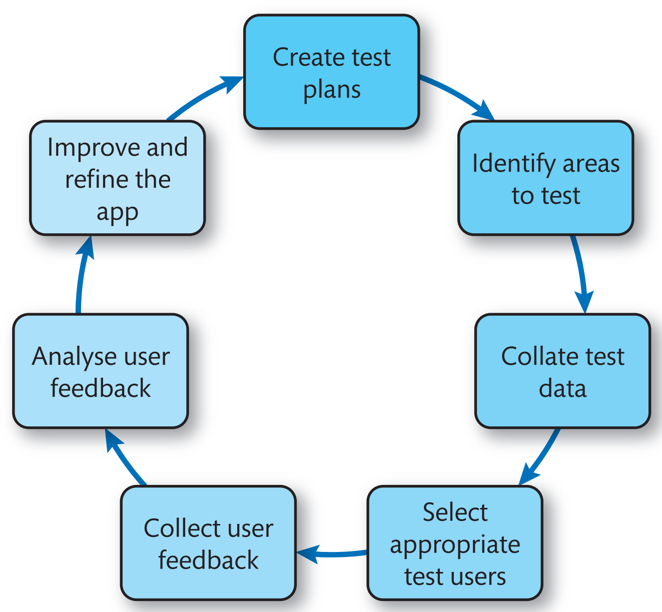


Figure 17.40: The cyclical stages involved in robust testing of a mobile app

Test plans should detail in a step-by-step fashion how you are planning to test your mobile app. This will include the identification of several 'use cases'. Each use case attempts to tell the story of a user's interaction with your app (successful and unsuccessful) and typically includes a combination of:

- test data being entered
- operations being performed
- the order of the operations being performed
- the user's responses to your app's prompts.

Test data typically can be found in one of three possible states.

- Normal – data within the acceptable range that the user would normally enter.
- Extreme – unlikely data at the edges of the acceptable range.
- Erroneous – data which should not be entered (eg text rather than numbers).

Another concept that you may encounter as part of the test plan is the idea of white and black box testing. White box testing is usually performed by the developer by tracing the use cases through the program code logic and completing **trace tables**. A black box test is performed by a user by following a use case. The user has no exposure to the program code and does not need to know how the app works, whereas you, as the developer, are only interested in the outcome that the user gets.

Areas which need to be tested include the following.

- Functionality – all functions in the app should work as expected.
- Acceptance – the app should be fit for purpose: that is, it should meet the identified user requirements.
- Installation – covers the app's initial installation (typically from an app store download), its updating and its eventual removal.
- Performance – how the well the app behaves and impacts upon the target device(s).
- Usability – the app should be intuitive to use and its interface quickly mastered by its users; getting a positive response from testers is often a key part of an app's success.
- Compatibility – the app should work consistently well across different models and brands of device.

Key term

Trace table – a table created by the developer, through white box testing, which charts the changes in program variables as a use case is performed. The trace table would have entries for each use case with expected (what should happen) and actual (what does happen) outcomes. Comparing these outcomes helps the developer assess whether the app is working correctly.

Link

For more on testing tables, see the section 'Test data' in *Unit 4 Software Design and Development Project*.

Test users and user feedback

It is advisable that you select test users who have had very little to do with the development process. This will ensure that their observations are honestly expressed and without the bias that project involvement typically brings. Users should be drawn from a random population sample, ideally with mixed levels of technical experience.

User feedback must be collected from the user. This may be achieved using a variety of fact-finding techniques including:

- direct observation
- one-to-one interviews
- focus groups
- questionnaires or surveys
- automated error reports generated by the app itself.

Collecting user feedback can be both time-consuming and expensive, so techniques have to be used wisely. A common (and cost-effective) technique is the logging of feedback from confirmed users using online app stores because it is free and unsolicited (for example, from Apple®'s App Store or Google Play™). The release of free Alpha or Beta (very early) releases of an app to selected users is also a useful tactic to consider.

Feedback should, ideally, be a mixture of quantitative and qualitative data which can be collated and analysed to produce useful information and identify patterns and trends in the user experience, which can highlight faults and areas for improvement.

Quantitative data – data that specifies quantities: that is, it can be measured and written as numbers, percentages or proportions. For example, '50 per cent of users experienced no issues while using the app'.

Qualitative data – are represented by reasons, opinions and motivations and can be used to explore (and make sense of) quantitative data. For example, 'users found the application difficult to navigate and text hard to read on smaller devices'.

Discussion

Look at the two examples of quantitative and qualitative results. Can a link be found? Is the reason why 50 per cent of users experienced issues when using the app (quantitative result) explained by the problems encountered with the navigation and text size (qualitative result)? Discuss in small groups.

Table 17.8 Shows quantitative and qualitative questions in a questionnaire.

Table 17.8: Quantitative and qualitative questions

Question 10a	How was MyApp's user interface?	☐ poor ☐ acceptable ☐ good ☐ outstanding
Question 10b	Please explain your reasons ...	

For exampe, if users' feedback suggested that while some found the user interface 'attractive and with a bright colour scheme' (qualitative data), you also find that '60 per cent of users judged the user interface 'poor' (quantitative data), then there is clearly a usability issue in your app's design.

If the supporting data is not available, it might be necessary to return to those users and investigate their unsatisfactory responses to find the underlying cause. Superior fact-finding should always ask for reasons and not just record a score.

Improving and refining your app

The process of improving and refining your app is guided by analysing the combined results of your testing (that is, done by the developer) and the received user feedback.

Issues to tackle should include:

- ensuring that the app has all the agreed functionality described in the design
- fixing run-time errors, bugs or application crashes to improve reliability
- fixing inaccurate calculations and outputs
- improving responsiveness
- improving user interface (e.g. colour scheme, typeface, font sizes, images)
- improving user navigation and app flow
- increasing speed/performance
- reducing the app's use of device resources by further optimising assets or simply programming more efficiently
- improving the compatibility of the app with different models/brands of mobile device
- adding any other desirable features or functionality which fall within the design specification.

Reflect

You will have reviewed your app using feedback from fellow professionals and test users. The views expressed by others are often very useful as they are typically unbiased, providing a neutral judgement about the design and implementation of your app. Learn to respond to outcomes and take this feedback in your stride. This will enable you to react in a mature fashion that leads to improvements in both the end product and your problem solving abilities.

Any changes made to the app should be recorded in a change log, which is a document written by the developer that details the fixes and improvements made for each version of the app. A version number is used to clearly identify each successive release. This uses a 'major.minor' system, for example:

- MyApp ver 1.0 – first major release
- MyApp ver 1.1 – minor update, fixing menu (from user feedback), improving compatibility
- MyApp ver 2.0 – major update, includes additional functionality.

As you can see, minor versions increment when small updates and fixes are committed (for example MyApp ver 1.1). Many minor releases may occur between major updates.

In contrast, major updates represent large changes to the app functionality, features or code (for example, MyApp ver 2.0). You may also have personal experience of later app releases introducing brand new errors and bugs. This is quite common and a hazard that every developer faces when trying to remedy an identified problem; in fixing a bug, they sometimes introduce others.

Lessons learned from developing a mobile app

It is important that you reflect and evaluate the effectiveness of the app that you have developed. This review encourages you to ask yourself the following questions.

- How well has the solution met the user requirements identified in the design document?
  Think about: how has your app met the targeted user requirements, which elements has it achieved and which elements are missing or incomplete?
- Which issues arose during testing and your refinement of the app?
  Think about: the issues you should have anticipated, how you could have prevented these problems from occurring, which coding techniques or device features could have been used better?
- How could the app be improved to meet the user requirements more closely?
  Think about: the testing and user feedback, the aspects that could be added or refined.
- With hindsight gained through your testing and user feedback, if you were to repeat the task, were there alternative solutions that should have been implemented instead?
  Think about: the testing, user feedback and the alternative designs in your design document.

Whatever the outcome, there is always room for improvement when developing a mobile app. One of the key parts of the learning process (and of developing better apps) is learning from your mistakes.

Reflect

Mobile app developers need to develop and demonstrate behaviours that fit the professional nature and expectations of the industry. Some key attributes you must adopt are:

- giving respect to others, especially their opinions when they are giving you essential feedback
- communicating appropriately with a target audience, providing and accepting open and honest ideas and opinions
- having an open mind to new ideas and alternative solutions and not being afraid to try new techniques to improve a solution or the target user's experience
- a commitment to personal improvement, by developing both new knowledge and practical skills
- the desire to achieve excellence through the standards of problem solving applied and the mobile apps created.

PAUSE POINT	Can you explain what the learning aim was about? What elements did you find easiest?
Hint	Describe the general steps of developing an app from a design.
Extend	What is the difference between quality control and formal testing?

Assessment practice 17.2

B.P3 B.P4 C.P5 C.P6 C.P7 B.M2 BC.D2 C.M3 BC.D3

You have been asked by your line manager to lead a development project for the design and implementation of a new mobile app that generates quotations for a landscape gardener. This involves selecting the prices of services that are involved (e.g. gardening, paving, planting, mowing) and the costs of goods (e.g. seeds, plants, bushes) that are to be included. Prices for the various services should be selected from local businesses although they should be easy to configure in the app itself. A standard rate of VAT is also expected.

Intending to demonstrate best practice to a newly recruited junior developer, you start by producing designs for a mobile app that meet the requirements you have identified. As part of your design process, you should review the design with your peers to identify opportunities for improvements or refinements. Before commencing development, you should justify and evaluate your design decisions to your junior colleague by explaining how they meet the identified requirements.

Once the design has been agreed, you must produce the mobile app from your design, testing it for functionality, usability, stability and performance. The working app should then be reviewed to ensure that it meets identified requirements and is optimised as necessary.

A final evaluation should be made of the optimised mobile app, comparing and contrasting it to the client requirements.

Plan

- What is the problem I am being asked to solve? Do I understand the client's needs?
- How confident do I feel in my own abilities to complete this task, in terms of design and coding?
- Are there any technical areas I think I may struggle with? If so, do I know how to research these?

Do

- I know what it is I am doing and what I want to achieve.
- I can identify through feedback and review when I have gone wrong and adjust my thinking/approach to get myself back on course.
- I am prepared to optimise my solution based on identified requirements and feedback.

Review

- I can explain what the task was and how I approached the task, justifying my decisions at all stages.
- I can explain how I would approach the hard elements differently next time (ie what I would do differently).
- I will use this as an opportunity to try new techniques and improve my problem solving.

Further reading and resources

Banga, C. and Weinhold, J. (2014). *Essential Mobile Interaction Design*. New Jersey: Addison Wesley.

Lee, W-M. (2012). *Beginning Android 4 Application Development*. New Jersey: John Wiley & Sons.

Burd, B, (2015). *Android Application Development All-in-One For Dummies*. New Jersey: John Wiley & Sons.

Lowe, D. (2014). *Java All-in-One For Dummies*. New Jersey: John Wiley & Sons.

Ray, J. (2015). *iOS 8 Application Development in 24 Hours, Sams Teach Yourself*.

Indianapolis: Sams Publishing.

Websites

Develop apps for Android™: **http://developer.android.com/**
This enables you to get started with Android Studio.
Develop apps for Apple®: **https://developer.apple.com/programs/**
The Apple Developer Program.
Mobile Development Tutorials: **www.tutorialspoint.com/**
Mobile Development Made Easy (for Android™, Apple® iOS etc.).

THINK FUTURE

Elizabeth Shaw

Junior Developer in a mobile app development team

I've been working on mobile apps for about two years and during this time I have been involved in developing many different apps, some for Android™ and others for Apple® devices. Although the technologies and languages are quite different, the design principles are generally very similar. Although sometimes my preferred design is pitch-perfect, this often isn't the case and I have to review the design with considerable input from my colleagues. I have to be mature about this and take suggestions and feedback in a positive way although often it will mean going back to the drawing board and trying other ideas.

Of course, sometimes I don't agree with my colleagues and I have to justify the decisions I have made during the design process. As long as I can prove that my ideas meet the identified requirements this is usually ok. Essentially, I am demonstrating my creativity and taking an individual responsibility for my efforts, something which my colleagues openly encourage.

Focusing your skills

Reviewing mobile app designs

It is important to be able to review the designs of a mobile app, both individually and with others, in order to ensure that they meet the identified requirements. Here are some questions to ask yourself to help you do this.

- What are the implications of implementing a poorly thought-out design?
- Which requirements need to be identified and addressed by your design?
- How can you justify the decisions that you made during the design process?
- How you would record the feedback you receive from others and how might you use it to improve the design?
- Finally, for each requirement you have identified, work out how you can ensure that it is dealt with in your design and review.

Designing a mobile app – questions on design requirements

The design for your mobile app has been reviewed by others. What happens next?

- If the mobile app design is poorly received and serious shortcomings have been identified in your understanding of the problem and/or your design, you will need to revisit the original client requirements.
- If the mobile app design is generally well received but issues have been identified with elements of your designs that rely on specific devices, platforms or the presence of integrated hardware, you may wish to refine your ideas.
- If the mobile app design is very well received, consider that there may still be room for improvement and seek advice and guidance from others who may have more practical experience than you.

Getting ready for assessment

Florence is working towards a BTEC National in Computing. She was given an assignment which asked her to give a presentation on 'mobile apps and mobile devices' for Learning Aim A. She had to cover all the different areas of mobile apps and mobile devices, with a particular emphasis on which aspects affect the design and implementation of mobile apps. Florence shares her experience of completing this assignment below.

How I got started

Having researched the different types of mobile app (including the different categories of app and whether they are native, hybrid or web apps), my next step was to think about how the users' needs determined the app's design. To do this, I asked a relative for an app that they would like on their mobile phone. I made notes of their needs (what they wanted it to actually do), their preferences (how they wanted it to look and work) and their characteristics (how they wanted to use it) and tried to link this to a potential design, explaining my decisions in terms of their requirements.

Another aspect I had to explore was how current technologies impacted on the design and implementation of mobile apps. I looked back at apps operating on various devices (for size comparison) and from differently aged models (older and newer phones).

I tried to analyse this further by selecting a number of different apps and comparing their design and implementation across various types and models of device. I managed to create a list of different aspects that enabled me to compare and contrast these really well.

In order to achieve the highest grade, I realised I had to make some judgements about the effectiveness of these apps. I conducted and documented a consumer test with my family and friends, asking them to compare a popular app across a number of different devices. They gave me a lot of feedback about the usability of each app and I collated this into a set of results which I used to rate each app. It was then possible to draw some conclusions about the users' needs, the purpose of the app and the technologies available in each device in order to evaluate how the app's overall design and effectiveness had been directly affected.

How I brought it all together

I typed my findings into a document and used this to create slides and notes which formed my presentation. I created some tables and charts to help me to explain my findings. This reduced the amount of text I needed to add and made the presentation look more interesting and professional.

I ensured that I had proofread each page carefully and practised the presentation with a family friend who pretended to be the target audience. This helped with my nerves and timings on each slide. When they became confused over some of the technical terms I had used, I simply changed them or added a quick explanation. At the end of the report, I added a set of URLs for the audience so that they could find out more information.

What I learned from the experience

I realised that, although my class notes provided a good starting point, it was essential to understand what I was being asked to do before I started work on the assignment. This proved to be good practice for developing mobile apps: always understand the problem fully before you start work on it.

I also learned that collecting other people's opinions was a very useful way of getting honest feedback and this helped me to rethink my own views on certain apps and mobile devices. Even when I didn't particularly agree with them, I still found their feedback interesting, something which will be useful when I receive feedback on my own app.

Think about it

- Do you know the difference between explaining, analysing and evaluating?
- Can you respect other people's feedback even when you do not necessarily agree with it?

Glossary

Act of God: an uncontrollable natural occurrence.

Arduino: this is a small, cheap hardware device containing a CPU and input/output (I/O) which can be used to explore programming and controlling devices.

Aesthetically pleasing: something which is visually engaging or appealing to the senses of sight and hearing. Usually such things are considered beautiful or attractive.

Alexa Traffic Rank: ranks websites based on a combined measure of page views and the number of visitors. From this, it creates a list of 'top websites' averaged over three month periods.

Algorithm: a solution to a programming problem that is used to help plan the code that will be written to solve the problem.

Alphabetical order: when data is placed in a sequence according to the order of letters in the alphabet which means that anything starting with the letter B would be sorted into place after anything starting with the letter A and before anything starting with the letter C. Using alphabetical order, number 12 would be ordered after 1 and before 2.

Android virtual device (AVD): a software emulated version of a particular physical device, running within a host operating system such as Microsoft Windows operating system or Apple OS X.

Animation cycles: different motions that characters will use on repeat such as running and jumping.

Animator: a job title that involves the creation of movement in game objects.

Anonymiser websites: these provide a method for bypassing proxy server filtering.

API: this stands for application program interface, which is the place where code from the app can send calls (requests) to the operating system to make the hardware do something such as opening a document.

Applet: a mini software application that can be built into a web page, for example a calculator that can be used by website users upon clicking a button on a web page.

Array: a collection of indexed variables, each of which has a single value.

ARP spoofing: ARP (address resolution protocol) is a protocol used in LANs to provide a mapping between the physical address of the devices (MAC addresses) and their IP addresses. An attacker can send fake (spoof) ARP messages onto a local area network (LAN) which associate the attacker's computer with the address of another device on the LAN, such as the default gateway or router. Therefore all messages being sent outside the network will be forwarded to the attacker's computer rather than the default gateway.

Assembler: this is a programming environment used for producing machine code programs. Use of low level programming is quite rare as most code is written in high level languages such as C# which are a lot easier to use.

Assumption: something that is accepted as true or as certain to happen without evidence. However, reasonable assumptions can be made based on prior experience and knowledge.

Asynchronous: computer systems each have clocks inside their circuits to providing timing pulses when moving data. Asynchronous signals do not need the clocks at each end of a data transmission to beat together.

Attributes: additional information about your wireframe. For example which font you will use, which font size you will use, where the side bar will appear, alignment and so on.

Audible: a sound such as speech, music or noise.

Backup: a copy of data files that is kept in case the original files are lost or damaged.

Benchmark: a point of reference that provides a measurement against which a comparison can be made.

Bespoke: software that is specifically written for a purpose as opposed to off the shelf software which can be purchased from a shop or downloaded.

Bitmaps: images that are made up of individual pixels that together display a picture.

Biometrics: the use of biological data for identification, for example voice control, facial recognition, fingerprints or iris scans. Biometrics uses features of a person that are unique to that individual. These bits of personal information are virtually impossible to forge. Biometrics also have an advantage over other identification methods because the individual does not have to keep anything (such as an ID card) or remember anything (such as a password). DNA identification technology is in its infancy but the idea of using a person's DNA to uniquely identify them has potential for future use in the IT security industry.

Black box testing: a testing method (also known as functionality testing) which tests all the functions of a program without the need for a peer inside the internal structure of the code.

Black hat SEO techniques: techniques that attempt to fool search engines into ranking a website higher than it would have otherwise been, by violating the search engines' terms of service and possibly by creating fake customer reviews.

Bluetooth: a short-range wireless standard that can connect a mobile phone to a headset or transfer data such as music between devices.

Botnet: a collection of zombie computers (very often home based computers) which are controlled remotely to carry out various actions such as sending spam emails or launching DDoS attacks.

Brand: an aspect of a product or service that distinguishes it from other similar products. It can include, among other things, a logo, colour scheme and name. Organisations often go to great lengths to develop a unique brand identity. Some of the best known brands in the world are McDonald's golden arches, Apple's symbolic logo and Nike's iconic swoosh/tick.

Briefing: an explanation of what is going to happen, such as the order of tasks or where the team members are located.

British Computer Society (BCS): the Chartered Institute for IT which collaborates with the Government and industry to set and promote common standards within IT.

Bug: a fault or error in a program which causes it to crash (end unexpectedly) or produce unexpected results.

Bureaucracy: excessive administrative procedures, often generating an overload of paperwork.

Business case: justification of a project according to the benefits it will bring to business functions and profits.

Bus: the motherboard has parallel metal tracks to connect different parts, allowing each bit in a byte to travel along a track side by side when transferring data between components such as the CPU and RAM.

Cache: this is the fast memory between a slow device and a quicker device that is used to prevent the quicker from slowing down when the slower device is used. Processors contain cache, which is used for command queuing, to help program code run faster when commands are brought in for the slower RAM to the quicker CPU.

CAD: computer-aided design software which is used for detailed technical drawings. It is used to design cars and buildings.

CAM: computer-aided manufacturing is done by machines which use software to operate tools and machinery for manufacturing, so providing greater precision and accuracy.

Capacitive touch screens: use two spaced layers of glass that are both coated with minute ITO capacitors. When the user's finger touches the screen, it changes the screen's local electrostatic field. This type of touch screen is brighter and more sensitive than resistive touch screens and supports multi gestures. However, the structural complexity of the screen raises its production costs.

Casting: where a value is converted from one type of data to another, potentially losing data during the process e.g. when truncating a decimal place, when converting a fractional number to an integer. Casting can be implicit or explicit.

Casual gamers: people who only play games for short periods of time and prefer simpler games.

Caveats: constraints, limitations, conditions.

Cel shaded: a type of non-photorealistic rendering. It is designed to make 3D graphics appear to be flat. It is often used to mimic the style of comic books.

Certificate authority: an organisation that issues digital certificates. There are many commercial CAs including Comodo, Symantec and GoDaddy. There is also a free CA called Let's encrypt which is sponsored by organisations such as Cisco and Google Chrome browser.

Chipset: the design of the circuits on a component such as a motherboard or a graphics card. Each release of a chipset is a new design that usually offers improved performance, often with faster operation and a lower power requirement.

Class: a term used in object-orientated programming whereby data and functions are collected together to form a class.

Click bait: a term used for content that encourages users to 'click through' the post to see the linked content on the external website. Click bait content often uses sensationalist headlines such as 'you won't believe what happens next' to exploit the viewer's curiosity.

Cloud storage: a virtual space for storing data provided through an internet connection. Advantages of this technology include being able to access the same data from almost anywhere, at any time on any device.

Collision detection: the process of checking to see which game objects have collided with each other.

Colloquialisms: words and phrases that are local to the region where you live and which are not grammatically accurate such as 'I should of...' which accurately stated is 'I should have...'.

Comments: explanations of the pseudocode or code. In programming terms, commenting is ignored by the program's compiler but allows a programmer to know what the code will do.

Compiler: a compiler is a special program that processes statements written in a particular programming language and turns them into machine language or 'code' which computers use to run programs.

Compression: the technique used to make files smaller. Compression can be a feature of a file format or can be implemented by utility software such as WinZip.

Concatenation: the process of joining things together. In programming it is used to describe the process of joining shorter strings together to form longer ones.

Concept art: drawings and paintings created before a game is developed to show how the game world should look and feel.

Confidentiality: principle of keeping something a secret. If you are told some information in confidence then you must keep it a secret.

Contingency: anticipated allowance made for unexpected future events; in this case additional funds for the project.

Continuous professional development (CPD): refers to any training or activities that an employee undertakes to keep their knowledge current or to enhance their skills.

Context-sensitive perspective: when the in-game camera changes to focus on something that the player has selected or a change that they have caused.

Copper: in this context this is the name given to cabling using any metal to conduct electrical signals.

Core: the part of the CPU that is able to run code. Up until 2005, all processors had single cores as it was not possible to make the circuits small enough to fit two cores into one chip before that. 'Quad' (4) core processors are now common but processors with higher numbers of cores do exist.

Corruption: in this context of data, this is when one or more bits get misunderstood by the receiving computer system during a data transmission.

Covert: undercover, such as a mystery shopper who assesses the quality of operations and performances of staff without the staff or shoppers being informed of when (and possibly if) this will take place.

Creditors: the accounting term given to those to whom a business owes money.

Crib sheet: a quick reference checklist of items to cover (in a telephone conversation for example).

Cut scene: a cinematic sequence in a game that tells part of the story. Cut scenes can be a separate movie clip or can be shown during a level.

CV: stands for curriculum vitae which is a chronological record of an individuals' experience, qualifications and employment details.

Cycle: instructions or operations which are repeated.

Cyclical: a process which is repeated and goes round and round in sequence.

Cypher: algorithm used to encrypt and decrypt data (alternative spelling is cipher).

Data types: used to specify the kind of information that the programmer needs to store. Booleans, integers and strings are just some examples of data types.

Debriefing: an event where the relevant stakeholders reflect on how the task or activities went in relation to the outcome and whether they met the deliverables.

Decomposed: when a complex problem is broken down into more manageable component parts.

Default setting: the setting selected by the games engine when nothing else has been input by the developer or, if the game is running, by the player.

Deficit: financial loss.

Demographics: measurements that are used to put people into different categories. One reason for this is to understand their likes and dislikes more easily.

Denial of service (DoS) attack: an attack on an organisation's website which involves sending so many bogus requests to the server where the website is hosted that it is overwhelmed and cannot respond to legitimate requests. The likely purpose of the attack is either revenge or blackmail.

Digital certificate: to run a secure website, the site owner must apply to a certificate authority to obtain a digital certificate. The digital certificate certifies that the website is genuine and owns a particular public key. Without a digital certificate it might be possible for a bogus website to impersonate a secure sight.

Direct advertising: where an organisation uses adverts which tell you to buy their product or sign up to their service. They use a direct approach by simply telling you what the product/service is and suggesting that you purchase it.

Dot operator: a full stop (.) used to define what method an object will use.

Dry run: a method used by code developers to carefully follow how a program will run, by keeping track of the contents of variables at every point and checking that the decisions are based upon the correct values so that the program follows the expected code.

Efficiency: a measurement of the useful output of a task or activity. A high efficiency means little wastage of time, materials or labour.

Engagement: the number of people who interact with a post in some way (like, comment or share it). Post reach is good but engagement is much better because it means that not only did people see your post but they found it interesting enough to interact with it in some way.

En-masse: a group or large body of people.

Estimating: carrying out a calculated assumption which is not a simple guess but is carefully considered, which results in the best possible approximate answer having taken all known facts into account.

Event handling: describes the process of using a specially written function or class method to perform set actions when the user or system event is triggered. For example, outputting the result of a calculation when a button is pressed or displaying a device's 'low battery' warning indicator when it falls below a certain charge level.

External benchmark: a baseline measure deemed reliable enough to use as a level of judging whether the performance of an operation is equal to, better or worse than the standard expected.

Expiration: in the retail industry, this is the date by which all stock must be removed from sale according to the sell by date.

False-positive: when an alarm is raised by an event which is non-threatening. Anti-virus or anti-malware software and firewall systems may generate false positives when they look for the type of activity that may indicate an attack. It is impossible to prevent false positives altogether, but too many of them can be annoying and may lead to alerts being ignored, which could allow a real attack to continue without action being taken.

Fetch-execute cycle: the sequence repeated thousands of millions of times every second by the CPU to fetch the next instruction code from memory and then run that code.

FIFO: First in, first out – a way of describing how data is treated in some data structures. The first item data added is also the first item of data that may be removed.

File handling: programming to access files and folders from the hard drive of a computer.

Filepath: the term given to the specific location of a document including the directory in which it is filled.

Filter: by using a filter you can include or exclude certain values when running a search.

Fit for purpose: a term used to ascertain whether the end product (in this case the software) meets the requirements of the client.

Formalise: turning an informal arrangement into a formal one (by following convention) perhaps by using documents or templates and charts. Gantt charts are often used in project planning.

First fix: the term given to all the structural work that takes place to get a building to the stage where cables pipes and so on are installed. This is before the walls are completed and plastered and the property is completed.

Framework: a particular set of development tools that can be accessed via its API.

Franchise: a series of game titles that feature the same world, the same characters or the same setting.

Frequently asked questions (FAQ): a common feature of websites and other documentation. These are simply a list of common queries that people make along with their solutions.

Function: a separate block of code which performs a specific job.

Function or procedure: a function is a block of code, ideally between 5 and 50 lines in lengths, which has a single defined purpose. Although written just once, it can be executed many times during a program by using a function 'call', reducing the need for repeated code. In some programming languages the terms 'function' and procedure are used interchangeably, but in others they were very different concepts; a function typically returns a calculated value while a procedure performs a single identifiable task.

Game development studio: a team of people who create computer games.

Game mechanics: the way a game works, its features and its rules such as double jumps or collecting items.

Game publisher: a company that releases games to shops or online platforms and pays for development.

Genre: a category of computer game that describes the style of play, types of challenges and the perspective of the player.

Global variables: these can be used anywhere in your program code.

Goes live: describes the first time a website has been uploaded to a web server and is made available to the public.

Go live: in the context of implementing an IT system, it means when a product or service is ready for use, i.e it is active.

Graphical nodes: colourful blocks used to represent the different functions of a programming language used to add easy interaction.

Grid: a network of lines that is projected onto a game level in order to split it into logical sections.

Hamming code error correction: This is used in RAID systems to utilise parity bits to rebuild corrupted data.

<head>: an HTML tag which is used to provide data about an HTML document (web page).

Hierarchy: to sequence actions in the correct order.

High-level program: this is code written within most programming environments. One command in a high level language for example pctResult.Image = Image.FromFile("../Pics/Correct.jpg"); would compile into many lines of binary machine code. High-level programming is much easier than low-level programming as much less needs to be typed and because the keywords are much easier to understand and use.

HTML: stands for hypertext markup language and is used to create web pages.

Human computer interaction (HCI): the study of how people interact with machines, and the best possible ways to design interfaces between people and machines.

Hypertext: text that contains links to other bits of text.

Identifier: a programmer-friendly name which represents a value stored in the computer's RAM. Before the use of identifiers a programmer would need to know the actual memory address of a value in order to access it.

Idle cycle: the animation that a character displays when the player is not moving them. A character in an idle cycle may stand still and breathe or tap their foot after a few seconds.

IEC: International Electrotechnical Commission.

Immersive: a term which refers to how focused you are on the experience that you are having. An immersive game will keep your attention for long periods of time and block out distractions. It should make you enjoy the game more, but only if you have the time to spend on it.

Indentation: starting text further from the margin than the main body of text.

Indirect advertising: more subtle than direct advertising. It attempts to create a positive attitude towards the product or service in the mind of the customer through sponsorship (e.g. an organisation sponsors a particular TV programme), product placement and other methods (such as those that social media uses) to try to create a relationship with the with the customer.

Infinite loop: a sequence of instructions that is continually repeated until a certain condition is reached. There is no exit from the infinite loop so it repeats indefinitely.

Integrity: principle of being honest and having strong moral standards.

Interactive website: involves some level of activity from a simple feedback form to a database that personalises the website for each individual visitor. Changes can be made to the website 'on-the-fly'.

Interatively: in software development, each part of the development process is separated into small parts and repeated until complete.

Internet protocol (IP): one of the most important protocols is the TCP/IP protocol suite which is used for communication in local area networks (LANs) and via the internet. IP addresses are assigned to all devices that are participating in a network that uses the internet protocol.

Internet relay chat (IRC): the technology behind instant messaging that enables you to write messages and receive them from your friends or colleagues while using a computer.

Inversion: means flipping a value. In a Boolean inversion 0s are inverted into 1s and 1s are inverted into 0s. In a mathematical inversion a fraction is turned upside down so 2 would invert to ½ and ¼ would invert to 4. This is also called finding the reciprocal of a number.

ISO: Internal Organisation for Standardisation.

Iteration: something that is repeated over a cycle or period often building upon each previous stage.

Keywords: words that identify the key things that an organisation has to offer to customers and are likely to be used by potential customers when carrying out an internet search for a product or service.

Lag: a delay or reduction in the game's frame rate.

Lessons learned: insights gained from an experience which can be usefully applied to future projects.

Library: a package of code that can be reused by programs many times over.

Life cycle: in software development, the cyclical stages required to produce new software.

LIFO: last in, first out – a way of describing how data is treated in some data structures; the last item of data pushed on is also the first item of data that may be pulled back off.

Linux: an operating system (OS) which is released 'open source' meaning that the source code that creates it can be downloaded and adapted by anyone. Due to its adaptability it is very popular and comes in lots of different versions such as Steam OS.

Literal: this is the value in an algorithm or program which can be numeric or string.

Local variables: these are limited to being used in the block of code in which they are declared, for example within a particular function.

Logical operations: these are carried out between binary numbers using AND, OR, NAND, NOR, XOR, and NOT logical operators.

Long term evolution (LTE) modem: mobile phones use long term evolution (LTE) modems to send and receive internet data.

Low level program: code written using instructions which may map very closely to the CPU's actual instruction set. The CPU instructions are just binary numbers. In low level programming these are replaced with mnemonic. For example the CPU instruction for the add instruction might be 01000000, in a low level program this might be replaced with the mnemonic ADD.

MAC address: a unique fixed hardware address that is permanently coded into a network device such as a network interface card, by its manufacturer. A MAC address is a 12-digit hexadecimal number. The allocation of MAC addresses is managed by IEEE (Institute of Electrical and Electronic Engineers).

Main (): any C ++ program must contain a function named main, which is the designed start of the program.

Malware: an umbrella term for software that has a malicious intent. Malware has a deliberately negative effect on a computer. Once installed, it can gather private information, slow the computer down, delete or lock access to files for example.

Mantissa and exponent: these are used to represent very large or very small numbers in a small space using floating point or scientific notation. The mantissa is the number part and the exponent shows how many places the point has moved, e.g. 23,467,334 in denary can be shown as 2.35×10^7, where 2.35 is the mantissa and 7 is the exponent.

Massively multiplayer online (MMO) game: a game played by multiple players across the internet all online at the same time.

Memory bus: is used to move data in a computer system between the CPU and the RAM.

Mirroring: RAID uses mirroring to save two copies of the data so if a drive fails a good copy of the data can be brought back from the mirror.

Mission-critical: a mission-critical system is one which if it failed or suffered from errors could cause serious problems such as disruption affecting a large number of people, significant financial loss or even loss of life. Examples of mission critical systems include air traffic control, vehicle control systems and medical systems.

Mnemonics: acronyms used in low level programming environment for producing machine code programs. Most machine code instructions are made from the opcode (command) and the operand (value or address the opcode is to use). Each mnemonic represents a single machine code opcode instruction such as DEC which will decrement the operand (decrease by 1).

3D Modeller: a job role that involves the creation and texturing of 3D objects.

Module: a part of a large software system that carries out a specific function. For example different departments will use different modules within a full IT system such as Human Resources department using a payroll module to calculate staff wages.

Modulus division: performs a division operation and returns the remainder rather than working out the decimal or fractional answer.

Motherboard: a circuit board within a PC that connects all the main components.

Motion capture: a process where the real life motion is converted into data that can allow game characters to move in the same way.

Namespace: the line of code using namespace std; is useful to declare at the start of a C++ program, because it means you can write just cout instead of std::cout. This will ultimately save you time and take up less code.

NAND flash memory: this kind of memory is used in SSDs, memory sticks and memory cards to hold documents, images and other files. This technology is non-volatile which means that it does not require power to store data. The name NAND is used to differentiate this technology from NOR flash memories. NAND and NOR are used to describe the algorithms used to store data rather than the type of logic gates used in their circuits.

Nested formulae: a formula (or several formulae) within a formula such as 'if...else' used for example to calculate the due date of outstanding payments.

Node: a point where lines intersect.

Non-repudiation: the assurance that something cannot be denied. Typically this involves using a digital signature to prove that a contract or email was agreed to or sent by someone and they cannot deny it.

Numerical order: when data is placed in a number sequence using in ascending order which means that 12 would be sorted into place after 11.

Obfuscation: to make something more difficult to understand that is to make it obscure, unclear or unintelligible.

Object: a type of data that knows things about itself and knows how to do things.

Object-orientated language: uses code that is organised into objects which can be used to make it run in a fast and robust manner.

Object-oriented programming: a programming methodology based on the concept of 'objects' which are structures that contain data.

Object-orientated programming language: a modern programming approach to software development that works by modelling real-world problems and simplifies complex processes into the basic interactions that exist between different objects: for example, a customer and their bank account.

On account: the term given to the period of time offered to businesses by suppliers before their invoice payment are due to be settled. It is also referred to as 'credit' although when an individual buys something on credit they are usually liable for interest payments whereas an 'on account' business arrangement does not incur interest if payment is made within an agreed period.

Operating system: software that manages computer hardware and software resources.

Operators: special symbols (or multiple symbols) which tell the program to perform a specific arithmetic, relational or logical operation on its data. Operators have to be used in a specific order of precedence (this describes the order of execution). You may be familiar with this concept from using BIDMAS or BODMASS in mathematics.

Optical: in this context this is cabling that uses thin glass fibres to pass information using flashes of light. Optical cabling usually transmits data faster than copper cabling.

Optimised: optimised assets are created using file formats which are efficient as they require less storage space. This can result in improvements in performance and a smaller digital footprint on the device's resources. Examples include using .jpg images rather than .bmp files as these use data compression to reduce file size.

Overt: a completely transparent and open operation where all the affected parties are well informed in advance.

Paging: a technique used by the PC or laptop to use the backing storage as an overflow area for RAM if the RAM fills up with programs and data which would slow the system down. The technique is called paging because RAM is treated as a number of same sized pages to make swapping in and out of disk easier.

Patch: a series of code fixes that are downloaded and applied to the game code in order to fix problems.

Parametric: where parameters are assumed so that an analysis can be more accurately be estimated.

Parity error correction: an error-checking technique by which an extra bit is placed at the end of each byte to make the number of ones in the byte (and parity bit) an even number. This is called even parity; a similar technique where the parity makes an odd number of ones is called odd parity. Double parity error checking is when RAID writes a second set of parity across the drives as an extra safeguard against data loss.

Physics: in the real world, gravity, mass and other laws of physics apply naturally, but in the world created by a computer programmer, the laws of physics have to be made to apply through coding the game correctly.

Photorealistic games: games that try to look as lifelike as possible.

Pixel perfect: a term used in the design sector to describe graphics that are accurate to the very last pixel.

Platform: can refer to the operating systems that is used on a system or to the method of delivery such as cloud based applications.

Plug-in: software that will play specific types of files. For example modern versions of web browsers like internet Explorer come with Flash Player which is a plug-in to allow the user to play Flash animations.

Pointer: a special type of variable which stores the memory address of another variable in the RAM. This allows the second variable to be accessed indirectly by referencing the first variable to be accessed indirectly by referencing the first variable. Pointers are used in more advanced programming to make solutions more efficient but can be dangerous if not used correctly.

Polymorphism: comes from the Greek word meaning 'many forms' and is a core feature of the object oriented (OO) programming paradigm. In C++, common uses of polymorphism include the overloading of functions and operations, so changing their use (but not their name) depending on the context in which they are used.

Populate: in computing or spreadsheet terms, to input data into an application such as worksheet, cell or record of a database.

Ports: a part of the design of the TCP/IP protocol suite that is the method of communication that the internet uses. IP addresses are used to identify individual computers, and ports are used to identify specific services or applications. For example, the HTTP protocol used to transfer web pages uses port 80.

Post reach: the number of people who see a particular post.

Pre-shared key (PSK): a cryptographic term which describes an encryption key that is provided to authorised users to allow them to encrypt and decrypt data.

Prime number: a number that can be divided by only itself and by 1. There are lots of prime numbers such as 3, 5, 7 and 151.

Project management: different methods and procedures used to keep a project on track and under budget by minimising and mitigating risks and issues.

Project kick off: the official launch of the project and the point at which the details of the project are promoted.

Pro rata: in proportion, sometimes calculated as a percentage.

Prototypes: small test game levels used to make sure that the key features of the game are working and to illustrate to clients and potential customers what a game will be like.

Pseudocode: a way of writing code without it being in a specific programming language.

Quality: the term given to a standard of measurement that can be graded against similar kinds when compared with each other or with a series of benchmarks.

Qualitative: narrative data such as conversations, text and observations. Qualitative data is difficult to analyse and the results are highly subjective unless verified using other methods to test the views for their reliability.

Qualitative questioning: questions do not have definite answers. It provides answers as to how or why and is used to gauge opinions and get more detailed feedback.

Quantitative: numerical data which can be analysed and used to present results in different ways for example percentages or proportions.

Quantitative questioning: questions have a definite answer, either a numerical value or specific answers in ranges. It focuses on statistical analysis.

Queen's English: spoken and written English language, the standard agreed by highly educated people in Britain.

RAID level stripping: RAID can be used to spread saving and reading data over the drives to improve speed. Conceptually this is thought of as a making strips from the data across the drive. Block level stripping divides the data into 512KB blocks with each stored onto one of the drives. Byte level stripping uses the same technique but divides the data into bytes. Similarly, bit level stripping divides the bytes into bits for storage.

Random number: a number in a list of numbers that is not related to the previous or to the next number in the list.

Ratio: a way of concisely showing the relationship between two quantities. A ratio is represented by separating the two quantities with a colon (:). For example a ratio of 1:2 is one quantity compared with something which is twice the first amount.

Real number: a real number can be a whole number, a rational number (a fraction or a decimal) or an irrational number. Infinity and imaginary numbers (the square root of minus 1) are not real numbers.

Recursion: an important concept in computer science and occurs when the success of solving a larger problem relies on a solving smaller copies of that problem first. In programming recursion is typically achieved when a function calls itself a number of times until a terminating condition is met and each result is returned to its parent function until the final result is attained.

Redundancy: compiled code within a computer program which is not necessary.

Regression testing: a type of software testing that seeks to uncover new software bugs known as regressions, in existing functional and non-functional areas of a system after changes such as enhancements, patches or configuration changes have been made to them.

Relational comparisons: are carried out between numbers or text using operators such as less than (<), greater than (>), equal to (=), less than or equal to (<=), greater than equal to (>=), and not equal to (<>).

Rendering: the process of converting game assets and environments into 2D images that can be displayed on a screen.

Rendering engine: the software in a games engine that converts virtual worlds into 2D images which are then animated.

Resistive touch screens: use two layers covered in an electrical conductive material. The two layers are kept apart until a finger or stylus press them together, which causes a localised change in the electrical resistance. This type of touch screen is cheaper to manufacture than capacitive touch screens but is not very sensitive and cannot support multi touch gestures.

Resource list: a list of staff, equipment and raw materials required for the project.

Responsive web design (RWD): makes your web pages appear correctly (look good) on all types of device, including desktop PCs, mobile and tablet devices.

RJ45: a common standard for network ethernet cable plugs and the socket or port that the cable uses to connect networked devices together.

Run-time error: a problem that occurs while an application is being used. These errors result in the application locking or crashing.

Sans serif: the word 'sans' is French, meaning 'without'. Therefore a sans serif font is one without embellishments at the ends of letters. An example of a sans serif font is Arial.

SATA: serial AT attachment (SATA) is the current standard for the cable connecting backing storage to the motherboard. This replaced the previous enhanced integrated drive electronics (EIDE) standard which was also known as ATA-2.

Schedule of works: a plan or procedure defining the activities that need to be carried out to achieve the project objectives. It is sometimes referred to as a statement of work.

Schematics: a technical diagram showing the content and function of game elements.

Scripting language: a programming language that requires a separate application to run such as a games engine or web browser.

Serif: a type of font that has embellishments at the ends of letters. An example of a serif font is Times New Roman.

Server virtualisation: it is very common for server computers to use virtualisation software that allows one large and powerful server computer to operate as if it were several smaller systems carrying out specific functions. In cloud computing this allows a data centre to provide a customer with what appears to be their own dedicated server computer but in reality it is just one of many different virtualised servers running on a single large physical server.

Service level agreement (SLA): a contract agreed between a service provider and the end user that specifies the standard and level of service expected and identifies how the quality of the service will be monitored and measured and by whom, how and when. SLAs are often used in IT support contracts.

Service oriented processing: the breaking down of complex problems into a collection of separate processes, each providing a specific service for client applications.

Set: a collection of distinct objects. Sets can contain anything and may consist of many different members.

SMART target: a specific, measurable, achievable, realistic and timebound objective of a project.

Snagging: the process of identifying and reporting outstanding items or areas that require attention in order for a project to be signed off as satisfactory.

Software as a service (SaaS): software as a service lets you use apps that are licensed by a remote provider through cloud technology.

Software development kit (SDK): a suite of software tools provided for developers to create an application for a specific platform, for example JavaScript SDK for Android. The SDK usually contains application programming interfaces (APIs), tools such as compilers and debuggers, reference documentation and code samples.

Specification: a description, often detailed, of the design of a hardware device. This specification gives details of the user interface that it uses, the processing power and the amount of RAM, for example.

Sprites: images used to represent characters and objects in a 2D game.

Static website: one with no interactivity which is usually just a presentation of information. Changes to the website have to be hard-coded into the website.

Statistics: the practice of gathering and analysing large amounts of numerical data such as that used in scientific research.

Status: the role an individual plays or their job title in a professional setting.

Steganography: the practice of concealing information within other non-secret data.

Sub-contractor: an individual or business paid to work on a project, usually known as a third party, and not an employee of the company that the project is being carried out for.

Switch: a box with a number of ports for network cables to plug into. The circuits inside the switch send any data they receive to the port holding the cable that connects to the device where the data is needed.

Symmetric and asymmetric key encryption: in symmetric encryption, the same key is used to both encrypt and decrypt the data, whereas in asymmetric encryption two keys are used, one to encrypt the data and another, different, but mathematically paired key, is used to decrypt the data.

Synchronised: computer systems each have clocks inside their circuits to provide timing pulses when moving data. Synchronised signals of a data are transmitted to beat together.

Syntax: a set of rules that is unique to each programming language which defines how commands and instructions should be structured and ordered.

System on a chip: a single integrated circuit chip which combines all the functions of the computer. They are used in mobile systems because of their small size and low power consumption.

Technical artist: someone who works between the technical and design teams and understands both.

Time driven processing: a form of event driven programming, whereby each process is triggered by a time-based event, typically using a computer's real time clock (RTC).

Trace table: a table created by the developer, through white box testing, which charts the changes in program variables as a use case is performed. The trace table would have entries for each case with expected and actual outcomes. Comparing these outcomes helps the developer assess whether the app is working correctly.

Transport layer security (TLS): a protocol that makes certain that there is privacy between communicating applications and their users on the internet. TLS is the successor to the secure sockets layer (SSL) method.

Trigger: invisible collisions in a level that will prompt trigger a function or event.

Two's complement: method used in binary arithmetic to approximately represent negative numbers. To convert a binary number into two's complement, firstly invert the number (swap 1s for 0s and 0s to 1s) then add 1 to this inverted number to calculate the two's complement. The same operation (invert then 1+) is used to convert a two's complement number back to binary.

USB: a universal serial bus is a widespread socket or port found on computers, televisions, DJ decks, satellite navigation systems and many other devices.

Validate: the process of ensuring that data or systems are accurate by using original sources back to their origin.

Validation: a process that checks to see if an input value makes sense and is reasonable before it is processed. Validation does not check to see if the data has been keyed in accurately, this is a separate process called verification.

Variable: used by program code to hold value as the program runs.

Verified: the process of ensuring that data and information are reliable and accurate.

Viral: a social media post is said to have gone 'viral' if it is shared by numerous people who themselves share it and so on. As more and more people share the post, the number of people who see it can increase into millions. Having a social media post go viral is often considered a good thing unless of course it is regarding a negative issue about an organisation.

Wireless access point (WAP): most homes have a wireless access point (WAP) built into the internet router so that the laptops and other devices can use WiFi to access the internet. Many organisations also have dedicated WAPs arranged so that WiFi is available for staff and visitors. A dedicated WAP is connected to the network and does not include a router. Companies providing WAP access to their network for employees' BYOD access use authentication software to ensure that only authorised employees can gain access to the network.

Wire-framing: an important step in the screen design process, helping the developer to plan layout and user navigation using paper-based or electronic models of the devices and their visual components.

World Wide Web Consortium (W3C): an international community that develops open standards for the use of HTML5 to ensure the long-term growth of the worldwide web.

Zombie: a computer which has been compromised by an attacker and used to carry out a variety of malicious attacks under the remote control of the attacker. The owner of the computer is unaware that this has happened.

Index

Note: **Bold** page numbers indicate where a definition of the term can be found.

Q

R

S